DIREÇÃO CONCURSOS

Aula 00

Bancos de Dados para Analista - Tecnologia da Informação do TCE/RJ

Prof. Arthur Mendonça

Sumário

Introdução

Nosso Curso

ATENÇÃO! Para compreender bem o assunto desse curso, recomendo que primeiro finalize o estudo da matéria de **Análise de Informações**. Se você deseja estudar ambas as matérias em paraelelo, é importante que estude pelo menos até a aula nº 02, que trata de bancos de dados relacionais. Seu entendimento será fundamental para a compreensão desta e das próximas aulas.

Hoje vamos abordar os seguintes tópicos:

1 Conceitos básicos. 1.2 Topologia típica de ambientes com alta disponibilidade e escalabilidade. 1.3 Balanceamento de carga, fail-over e replicação de estado.

Iremos começar falando brevemente sobre o armazenamento de dados nas camadas físicas dos bancos de dados, para em seguida começar a detalhar técnicas de replicação, RAID e sistemas distribuídos de modo geral, pois esse é o tipo de topologia adotado para ambientes que requerem alta disponibilidade e escalabilidade.

Vamos lá?

@analisedeinformacoes

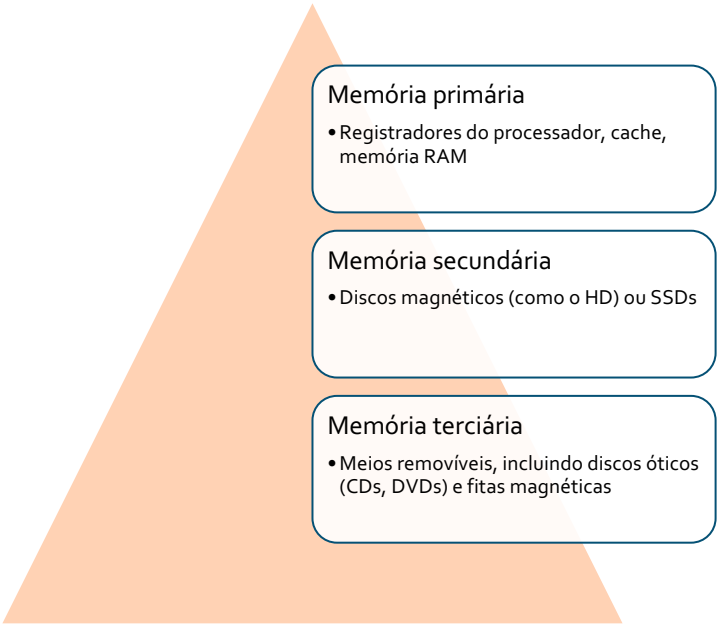
Teoria da aula

Armazenamento

Se você já estudou o nosso curso de **Análise de Dados e Informações**, você já estudou bastante a respeito da organização lógica dos dados no banco de dados. Você já sabe o que é uma **relação**, o que são **registros** e também **atributos**. No entanto, apesar de termos falado muito em modelos e conceitos, ainda não estudamos a respeito dos meios de armazenamento físico utilizados para os bancos de dados.

Um ambiente de banco de dados gerenciado por um SGBD é um sistema de computador similar a qualquer outro, já que é baseado em **software** rodando em um **hardware**. Ou seja, um sistema rodando em um computador. Como qualquer computador, os servidores de bancos de dados armazenam dados como **bits** (valores 0 e 1 que representam a unidade mais básica da informação) em dispositivos como um **disco rígido** ou um **SSD** *(solid state drive)*.

Esses meios de armazenamento são chamados **memória** ou simplesmente **armazenamento** *(storage)*, estando organizados em níveis hierárquicos:



A **memória primária** (ou armazenamento primário) é um tipo de memória que pode ser acessada **diretamente pelo processador (CPU)**. Por esse motivo, ela deve ser uma memória de **acesso extremamente rápido**, pois não adianta nada ter um CPU muito potente se ele precisa ficar esperando a memória responder às requisições de dados.

Já a **memória secundária** é composta pelo **HD** ou **SSD** do dispositivo, que são os meios de armazenamento permanentes que guardam a maior parte dos dados de um sistema. Além dela, existe a **memória terciária**, que diz respeito aos meios removíveis de armazenamento, que, como vamos ver, são comumente utilizados para dados de **arquivo** ou **backups**.

Professor, mas por que essa hierarquia? Por que não utilizamos só um tipo de memória pra simplificar? Boa pergunta! Essa hierarquia faz sentido, uma vez que os sistemas têm diferentes necessidades de velocidade e tamanho de armazenamento para diferentes níveis e operações realizadas.

Maior velocidade
Menor capacidade
Maior custo por bit

Menor velocidade
Maior capacidade
Menor custo por bit

A memória primária, como já dissemos, precisa ter um acesso **extremamente rápido**, já que serve diretamente ao CPU para a realização das operações de processamento. No entanto, a tecnologia atual não permite que esse tipo de memória seja **grande o suficiente** para o armazenamento de dezenas de milhares de arquivos. Essa memória tende a ser

Para esses propósitos de armazenamento em mais larga escala, se utiliza meios secundários, como HDs e SSDs. Esses meios fornecem **mais espaço** a um **menor custo** que a memória primária, porém sob a pena de um **acesso mais lento**. Por fim, os meios de armazenamento removíveis são ainda mais baratos, mas têm um tempo de acesso ainda maior.

Dentro dos próprios níveis hierárquicos, temos diferenças de preço e capacidade. No armazenamento primário, por exemplo, os registradores do processador têm uma capacidade muito pequena, na ordem dos bits. O cache já é maior, na ordem de kilobytes ou alguns megabytes. Por fim, a memória principal (RAM) costuma chegar a centenas de gigabytes para as aplicações mais exigentes de bancos de dados.

Já em relação ao armazenamento secundário, os SSDs *(solid state drives)* vêm caindo de preço e gradualmente substituindo os discos magnéticos tradicionais (HDs ou HDDs), seja em aplicações domésticas ou, em alguns casos, em **bancos de dados**. Os SSDs são dispositivos de memória *flash* não volátil, similar à utilizada em cartões de memória ou no armazenamento interno do seu celular. Esses equipamentos possuem velocidades consideravelmente maiores que os discos rígidos, mas ainda possuem custo relativamente elevado para justificar uma substituição total dos HDs.

Elmasri & Navathe (2010) apontam que a maioria dos bancos de dados é armazenada em **armazenamento secundário**, como em discos magnéticos (HDs), pelos seguintes motivos:

- Geralmente, os bancos de dados são **muito grandes** para caber na memória principal (a memória RAM).
- O armazenamento secundário tem ocorrência menos frequente de **perda de dados**, sendo por isso chamado, juntamente com o armazenamento terciário, de **armazenamento não volátil** (persistente).
    - O armazenamento da memória principal, em contraste, é chamado de **armazenamento volátil**, pois está mais sujeito a perdas.
- O **custo** de armazenamento é muito menor para o armazenamento secundário do que para o primário.

Como vimos, o CPU tem acesso direto somente à memória principal. Para que os dados armazenados pelo banco de dados sejam processados, eles precisam ser retirados do armazenamento secundário em que se encontram e **transferidos** para a memória principal. Esses dados são transferidos em **blocos ou páginas**, pequenas

porções do disco cujo tamanho é definido pelo **sistema operacional** no momento da formatação ou inicialização do disco.

Esses blocos podem estar distribuídos de uma forma **não sequencial** no disco rígido. Para conseguir saber onde cada bloco de dados se encontra, cada um deles deve ter um **endereço**, utilizado para acessá-lo e transferi-lo para a memória principal. Lembre-se, no entanto, que dois blocos que guardem dados similares não necessariamente estarão armazenados de forma contígua (consecutiva) no disco.

Quando vão ser realizadas operações de processamento sobre os dados, os blocos são transferidos do armazenamento secundário para uma **área reservada da memória principal** chamada ***buffer***. Os dados são trabalhados pelo processador e transferidos de volta do buffer para o disco quando as operações terminaram de ser realizadas.

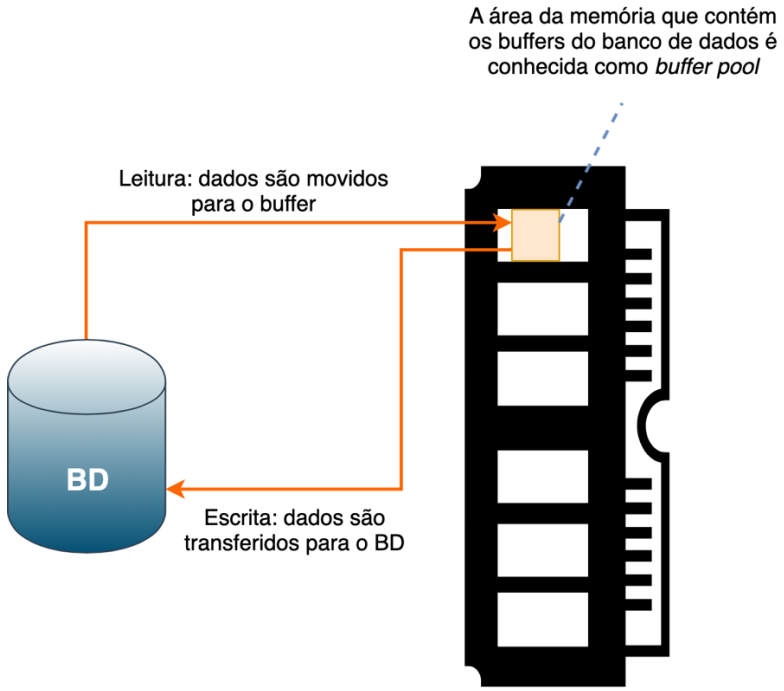


Figura: Diagrama simplificado representando as transferências de dados entre o BD (armazenamento secundário) e o buffer

É possível a transferência de múltiplos blocos por vez, caso em que o tamanho do buffer será ampliado para receber esse ***cluster*** de blocos. É possível a reserva de vários buffers na memória principal para a transferência simultânea de vários blocos. Isso é bem interessante no caso em que há a possibilidade de **pararelismo** ou mesmo a mera execução **intercalada** de transações com vários buffers ao mesmo tempo.

Armazenamento de registros

Em um SGBD relacional, os dados estão organizados em **registros** de tabelas, ou, mais formalmente, em tuplas de relações. Como você sabe, um registro é nada mais nada menos que um **conjunto de valores** para os atributos que compõem a estrutura da tabela.

Cada um desses atributos possui um um **tipo de dados**, como inteiro (int/integer), cadeia de caracteres de tamanho fixo ou variável (char ou varchar), booleano e assim sucessivamente. Cada um desses tipos de dados ocupa um determinado tamanho na estrutura de armazenamento. Por esse motivo, por exemplo, costumam existir vários subtipos para números inteiros, cada um ocupando um determinado tamanho na base e tendo uma capacidade máxima para a representação de valores. Veja o exemplo do Microsoft SQL Server:

Tipo de dados	Intervalo	Armazenamento
bigint	-2^63 (-9.223.372.036.854.775.808) a 2^63-1 (9.223.372.036.854.775.807)	8 bytes
int	-2^31 (-2.147.483.648) a 2^31-1 (2.147.483.647)	4 bytes
smallint	-2^15 (-32.768) a 2^15-1 (32.767)	2 bytes
tinyint	0 a 255	1 byte

Figura: Tipos de números inteiros e seus tamanhos no SQL Server. Fonte: Microsoft[i].

Os SGBDs modernos costumam oferecer suporte para o armazenamento de itens de dados **não estruturados**, como arquivos PDF ou multimídia (imagens, áudio ou vídeo). Esses itens geralmente são armazenados como **BLOBs** (*binary large objects*, ou objetos binários grandes). Elmasri & Navathe chamam a atenção que esses objetos costumam ser armazenados de forma **separada** dos registros. O registro da tabela que contém um BLOB, assim, mantém somente um ponteiro que referencia a posição do BLOB na estrutura de armazenamento.

Os registros estão armazenados em **arquivos**, que são sequências de registros. Os arquivos podem ser de tamanho fixo, caso em que todos os seus registros possuem o mesmo tamanho, ou podem ser de tamanho variável. O tamanho desses arquivos de dados costuma variar pela presença de **campos de tipo de tamanho variável** (por exemplo, o varchar), campos de **preenchimento opcional** ou mesmo pela presença de campos de **tabelas distintas** relacionadas no mesmo arquivo.

Esses registros nos arquivos de dados são armazenados nos **blocos** do meio de armazenamento utilizado. Um bloco pode conter uma série de registros, mas duas situações excepcionais podem acontecer: os registros podem ser **maiores** que o bloco de dados, ou pode **sobrar um espaço** menor que o tamanho de um novo registro ao final do bloco. Nesses casos, pode se usar **registros espalhados** *(spanned records)*, que são registros que se estendem por mais de um bloco. Isso se dá através de um **ponteiro** no final do primeiro bloco, que aponta para o início do bloco que contém a continuação do registro.

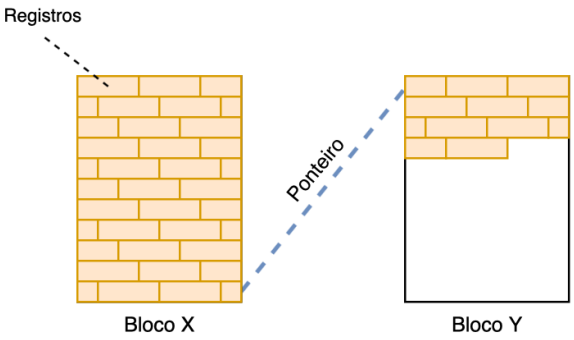


Figura: Registros espalhados em blocos de dados

RAID

Nas últimas décadas, a performance dos processadores cresceu em um ritmo mais rápido que a velocidade de acesso à memória secundária, utilizada como principal meio de armazenamento persistente de grandes volumes de dados nos sistemas de computador (Tanenbaum & Austin, 2012).

Para mitigar os problemas associados a esse descompasso entre velocidade de processamento e velocidade de acesso aos dados, pesquisadores decidiram aplicar a técnica de **paralelismo**, já vista nos processadores, aos dispositivos de entrada e saída de dados. Assim nasceu o conceito de **RAID** *(Redundant Array of Inexpensive/Independent Disks)*.

A ideia aqui é utilizar um conjunto de discos redundantes de modo a aumentar a **performance** de acesso com o uso de paralelismo ou mesmo a **confiabilidade** do sistema, por aumentar a tolerância a falhas nos discos individuais.

Com RAID, pode-se melhorar a **confiabilidade** e a segurança de dados utilizando uma técnica chamada ***mirroring*** (espelhamento) ou ***shadowing*** (sombreamento). Ela consiste em escrever os dados de forma **redundante** em discos distintos, ou seja, o mesmo conjunto de dados estará presente nos múltiplos discos que compõem o *array*. Isso tudo deve funcionar de maneira **transparente** aos usuários e aplicações, que enxergarão o conjunto de discos como se fosse um só. Caso um dos discos falhe, os dados podem ser lidos de outro disco no *array*.

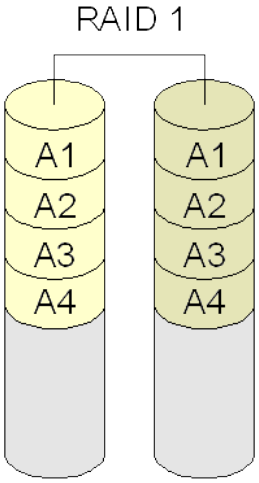


Figura: dados distribuídos em dois discos através de *mirroring*

Outra técnica utilizada para aumentar essa confiabilidade é o armazenamento de informações adicionais que não fazem parte do conjunto de dados original. Por exemplo, pode-se usar a técnica de **paridade**, que consiste em utilizar **um disco redundante** que possui algo como a **soma** dos outros discos. Os discos virtuais contêm, cada um, uma porção diferente dos dados. Assim, se um dos discos falhar, pode-se utilizar uma operação análoga à subtração para obter o dado indisponível:

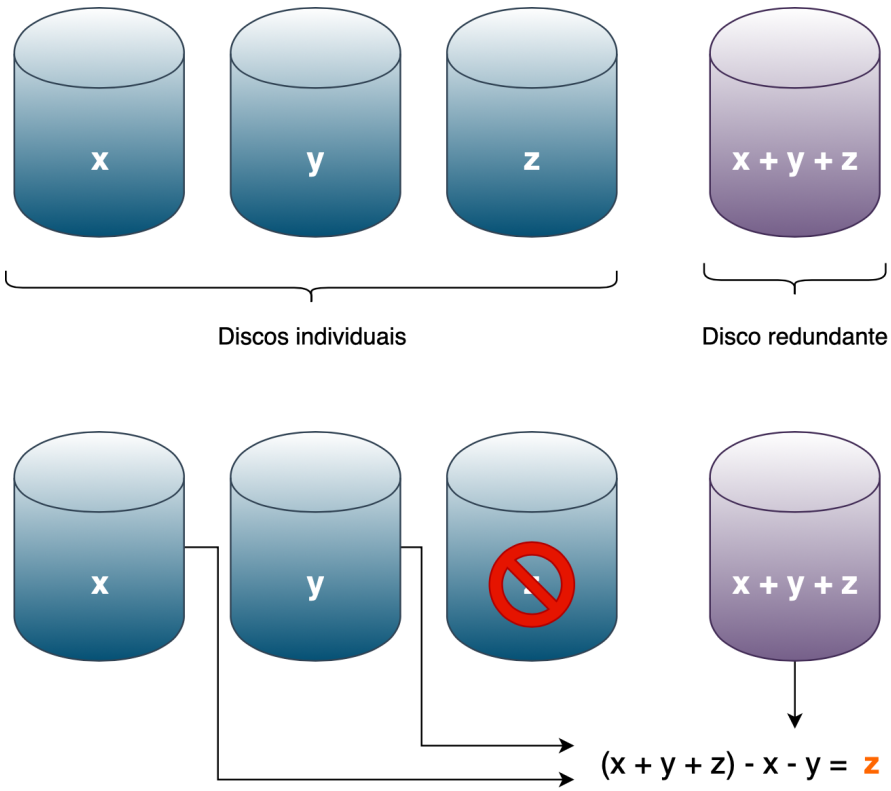


Figura: exemplo de esquema de paridade

O RAID também pode melhorar a **performance** ou velocidade do acesso aos dados através de uma técnica conhecida como ***striping*** ou distribuição. Essa técnica consiste em dividir os dados em faixas *(stripes)*, que são segmentos dos dados, em múltiplos discos. Assim, para um conjunto de dados de tamanho **x** dividido em quatro discos, teríamos que cada um guardaria uma porção de dados de tamanho **x/4**.

A granularidade das faixas pode variar. Podemos ter desde **bits** individuais sendo armazenados de forma sequencial, um bit em cada disco, ou mesmo a repartição dos dados na ordem de **blocos** inteiros de arquivos divididos ao longo do array de discos (Elmasri & Navathe, 2010).

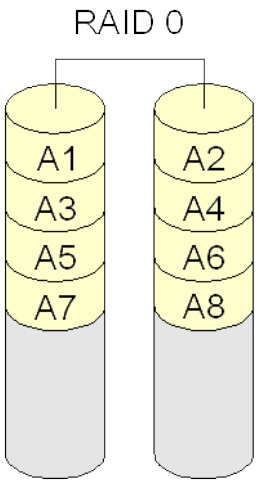


Figura: exemplo de *striping*

Dessa maneira, requisições de grandes volumes de dados podem extrair um pedaço de cada um dos discos do array, possibilitando uma leitura **em paralelo** das informações. No entanto, à medida que aumentamos o paralelismo, veja que também vamos diminuindo a confiabilidade do sistema, já que se somente um dos discos do array falhar, os outros discos não terão cópias dos dados indisponíveis.

Há várias organizações RAID possíveis, que são dispostas numa classificação que geralmente vai do **nível 0** ao **nível 6**. Essas organizações utilizam diferentes técnicas que mencionamos e mais alguns mecanismos mais sofisticados para melhorar a segurança e o desempenho do acesso aos dados. RAID é um conceito comumente associado a sistemas de computador e armazenamento de dados de modo geral, e que não costuma ser cobrado na parte de bancos de dados. Assim, considero que o que vimos até aqui suficiente para nossos propósitos.

(CESPE – SERPRO – 2008)

O espelhamento (mirroring), usado para a introdução de redundância, é uma técnica que aumenta a confiabilidade e por meio da qual os dados são escritos de modo redundante em mais de um disco físico e tratados como um só disco lógico. O armazenamento de dados que possibilitem reconstruir dados perdidos em caso de falha do disco — por exemplo, usando-se códigos para a correção de erros — é outra técnica para aumentar a confiabilidade.

RESOLUÇÃO:

Na utilização de múltiplos discos no RAID, algumas técnicas podem aumentar a confiabilidade. Uma delas é o espelhamento, caso em que cópias dos dados são armazenados nos diversos discos, de forma transparente para o usuário, que enxerga tudo como se fosse uma coisa só. Outra técnica possível é a **paridade**, em que armazenamos uma informação redundante que permite recuperar dados perdidos em caso de falha. Questão correta!

Gabarito: C

Bancos de dados distribuídos

Se você já estudou os conceitos de concorrência e paralelismo, sabe que uma alternativa para melhorar a performance de um computador é utilizando **múltiplos computadores**! Dispositivos executando em paralelo oferecem a possibilidade de se multiplicar o poder computacional do sistema como um todo ao distribuir a carga de trabalho em diferentes processadores, que resolvem os problemas de computador de forma simultânea.

No processamento das transações de um banco de dados, a utilização de paralelismo e concorrência de modo geral é bastante comum, dadas as necessidades de aproveitar ao máximo os recursos computacionais disponíveis.

Nos bancos de dados **distribuídos**, o conceito de se utilizar múltiplos dispositivos ainda existe, mas o propósito é diferente. Os sistemas paralelos consistem em **um só sistema** formado por múltiplos processadores que operam de forma conjunta, compartilhando os mesmos recursos como memória e dispositivos de entrada e saída.

Os sistemas distribuídos, por sua vez, consistem em máquinas **individuais** conectadas por uma **rede de computadores** e que realizam **cooperação** na realização de tarefas (Elmasri & Navathe, 2010). Esses sistemas em cooperação podem estar localizados em um mesmo local, numa mesma máquina (através de técnicas de virtualização) ou mesmo em localidades geográficas distintas. Assim, o sistema pode até estar **fisicamente espalhado**, mas estará **logicamente conectado**.

Essas máquinas podem ter configurações idênticas, caso em que o sistema será **homogêneo**. No entanto, isso não é obrigatório. Existem sistemas distribuídos **heterogêneos**, que são aqueles compostos por dispositivos com características distintas.

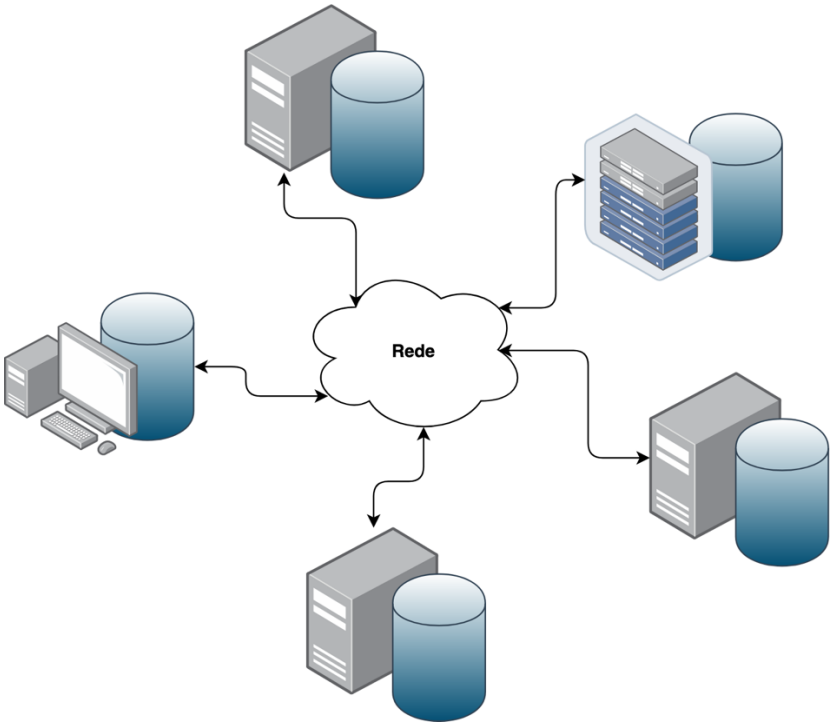


Figura: Representação de um sistema distribuído heterogêneo

Dessa forma, podemos resumir as características marcantes de um sistema de banco de dados distribuídos da seguinte maneira (Elmasri & Navathe, 2010):

- **Conexão dos nós de banco de dados através de uma rede de computadores**

Numa organização distribuída, os computadores que compõem a rede são chamados de **nós** ou **sítios**. Eles deverão estar conectados através de uma rede de computadores para transferência de dados e comandos.

- **Interrelação lógica entre os bancos de dados conectados**

Apesar de muitas vezes estarem em localidades distintas, os dados nos bancos de dados distribuídos devem estar **logicamente relacionados**. Vamos ver mais sobre como os dados são distribuídos mais à frente.

- **Ausência de necessidade de homogeneidade dos nós**

As máquinas que compõem os nós dos sistemas distribuídos podem ter configurações distintas de hardware e software e até mesmo conjuntos de dados distintos. Quando isso acontece, esses sistemas são chamados de **heterogêneos**.

Algumas bancas já cobraram questões relativas à performance de rede dos sistemas de bancos de dados distribuídos. É importante que você saiba que a organização de rede (topologia) utilizada, a distância entre os nós, a localização geográfica do terminal de acesso, dentre outras questões, **podem sim** influenciar na velocidade do acesso aos dados.

Afinal, os dados em uma rede de computador precisam trafegar através de meios como cabos, ondas de rádio e transmissão via satélite. Cada meio de transporte e cada arquitetura de rede de modo geral tem suas próprias particularidades e permite diferentes velocidades e latências. Não iremos entrar nesses detalhes, já que é um assunto da área de redes de computadores, não de bancos de dados!

Propriedades

Um sistema de bancos de dados distribuídos tem algumas propriedades que são frequentemente cobradas em concursos. Essas propriedades irão ajudar a entendermos melhor as vantagens e desvantagens de se utilizar um sistema desse tipo.

Confiabilidade e disponibilidade

Na aula de hoje, já falamos muito a respeito de **disponibilidade** e **confiabilidade**. É importante, no entanto, fazer a distinção formal entre os dois conceitos:

- A **confiabilidade** ***(reliability)*** diz respeito à propriedade de um sistema estar acessível e operacional quando for acionado, produzindo os resultados esperados. Assim, se você precisa do sistema e ele funciona, você pode **confiar** nele!
- Já a **disponibilidade** ***(availability)*** diz respeito à capacidade do sistema de funcionar corretamente em determinadas condições em um determinado período no tempo. Ela tem a ver com a **continuidade** do sistema. Assim, um sistema estará **disponível** quando está em condições de ser utilizado normalmente durante um determinado período.

É uma diferença sutil, já que ambos os conceitos estão associados à **não ocorrência de falhas** que comprometam o estado do sistema!

(CESPE – ABIN – 2018)

Julgue o item a seguir, relativo aos ambientes de alta disponibilidade.

Em um ambiente de alta disponibilidade, a confiabilidade é o resultado do levantamento estatístico das horas de funcionamento da infraestrutura elétrica do ambiente.

RESOLUÇÃO:

A medida descrita tem a ver com a **disponibilidade**, já que está falando em um levantamento estatístico ao longo de um período de tempo. De todo modo, nem mesmo a disponibilidade se resumiria ao mero funcionamento correto da infraestrutura elétrica do ambiente. O fato de o ambiente estar ligado na energia não necessariamente garante que ele está também em condições de uso normal, podendo receber requisições e respondê-las de forma adequada.

Gabarito: E

Transparência

A transparência em sistemas distribuídos, ao contrário do significado que costuma a ser atribuído no contexto da administração pública, é a propriedade do usuário **não enxergar** os detalhes da organização e da implementação dos sistemas distribuídos. Ou seja, para o usuário, não interessa a estrutura do sistema, o que ele quer é somente acessar os dados!

Elmasri & Navathe listam alguns **tipos** de transparência possíveis nos sistemas de bancos de dados distribuídos:

- **Transparência de organização de dados**

Pode ser transparência de **localização**, em que o comando que o usuário executa deve ser independente da localização do dos dados (o sítio em que eles se encontram) ou de onde foi executado o comando – essa costuma ser cobrada em concursos!

Também pode ser a transparência de **nomeação**, que se refere à capacidade dos usuários de acessarem os objetos do banco de dados através de seu nome, sem precisar especificar a localidade dos dados.

- **Transparência de replicação**

A replicação diz respeito ao armazenamento de cópias dos dados em diferentes sítios/nós. Isso pode ocorrer para aumentar a disponibilidade, a performance e a confiabilidade do sistema. A transparência de replicação, como se pode presumir, diz respeito à capacidade do usuário trabalhar com os dados sem precisar lidar com o fato de que existem diferentes cópias.

- **Transparência de fragmentação**

A fragmentação diz respeito à divisão das relações entre os múltiplos nós distribuídos. Os usuários não precisam tomar conhecimento de como uma relação foi fragmentada.

- **Outras formas de transparência, como transparência de design e de execução**

Design: conhecimento da arquitetura do sistema

Execução: conhecimento de onde uma determinada transação executa

Autonomia

É a característica que diz respeito à capacidade de cada nó do sistema distribuído de operar de forma **independente** dos demais. Os nós devem operar de forma coordenada, mas um certo grau de autonomia pode permitir que cada máquina seja gerenciada de forma individualizada.

Nesse contexto, podemos apresentar os **sistemas federados**, que são aqueles em que os sistemas **mantêm sua autonomia**, ou seja, eles podem continuar operando individualmente, com seus próprios usuários, transações, etc., mas ao mesmo tempo participam de um sistema distribuído, a federação (Sheth & Larson, 1990). É como se fosse a **União Europeia**: os países membros estão submetidos a um conjunto de regras em comum, mas cada um tem um alto grau de autonomia para lidar com seus assuntos internos.

Os sistemas federados são, em geral, **heterogêneos**, já que possuem seus próprios conjuntos de usuários e transações. No entanto, para que se configure a existência da federação, é necessário que haja um **esquema ou visão global** compartilhado por esses bancos de dados autônomos (Elmasri & Navathe, 2010).

(FCC – DPE/RS – 2017)

Quando se faz o particionamento de tabelas de um banco de dados relacional, por exemplo, para distribuir os dados, é característica que os usuários não necessitem ter conhecimento de quais computadores estejam armazenando as partições de dados. Tal característica recebe a denominação de transparência de

a) fragmentação.

b) criptografia.

c) local.

d) segurança.

e) replicação.

RESOLUÇÃO:

Tem pegadinha na questão! O que o examinador quer saber é qual o tipo de transparência que determina que o usuário não precisa saber qual computador (nó) está armazenando determinada partição. Isso é a transparência de **localização**!

A transparência de replicação, que você pode ter achado que era a resposta, diz respeito à ocultação de que determinado recurso possui várias cópias e de detalhes a respeito de como isso é realizado, não estando exatamente relacionada ao local de cada cópia dos dados.

Gabarito: C

(CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Um sistema de bancos de dados distribuídos consiste em sítios fortemente acoplados, que compartilham tanto a memória primária quanto dispositivos de armazenamento secundário.

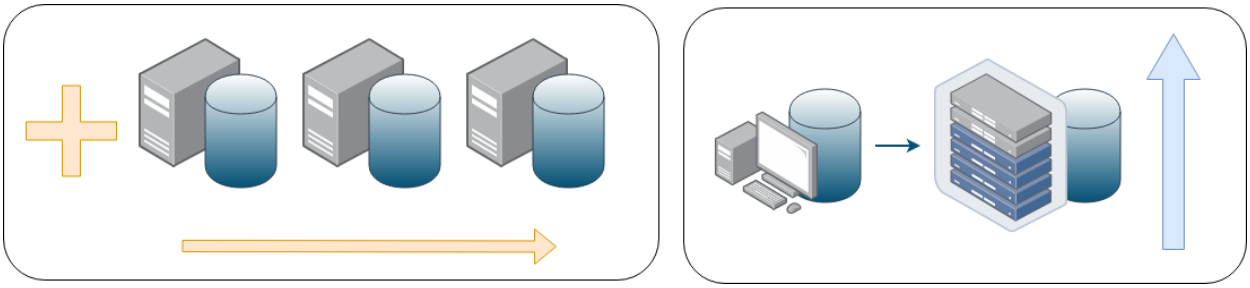
RESOLUÇÃO:

A descrição da assertiva é de um sistema de banco de dados **multiprocessador**, que possui várias CPUs que compartilham recursos de memória, formando um só sítio. No caso dos sistemas distribuídos, há uma distribuição dos dados em vários nós/sítios, que são máquinas independentes que **cooperam** na realização de tarefas.

Gabarito: E

Escalabilidade e Tolerância a Particionamento

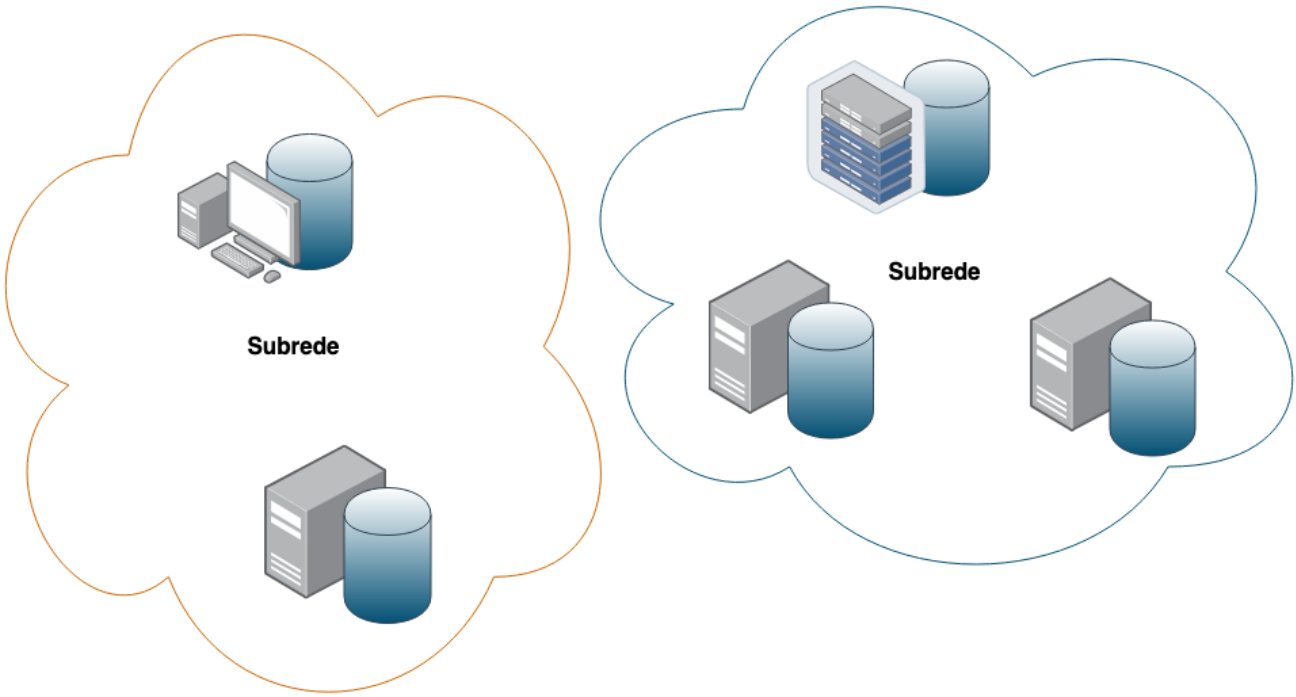
Essas duas propriedades estão bastante associadas aos sistemas distribuídos. A **escalabilidade** diz respeito à capacidade do sistema ser expandido sem que haja interrupção ou falha. Elmasri & Navathe apontam que há dois tipos de escalabilidade:



1. **Escalabilidade horizontal:** Diz respeito ao aumento da **quantidade de nós** no sistema distribuído. Um aumento nessa quantidade permite distribuir melhor a carga atual do sistema, seja em quantidade de dados, seja em capacidade de processamento. É como se um chefe contratasse mais um funcionário para uma equipe para fazer a mesma quantidade de trabalho – os funcionários antigos certamente ficariam menos sobrecarregados!

2. **Escalabilidade vertical:** Esse conceito, por sua vez, trata de expandir a capacidade de um nó atual do sistema. É o que ocorre quando um nó antigo, com um processador mais fraquinho ou com um disco de pouca capacidade, é substituído por uma máquina mais nova e potente. Há também um aumento da capacidade geral do sistema distribuído, mas sem ter havido um aumento da quantidade de nós.

Expandir um sistema distribuído de forma horizontal pode ter efeitos **indesejados**. Quanto mais nós, maior a sobrecarga de comunicação na rede estabelecida entre esses nós. Alguma falha pode acabar ocasionando um **particionamento** na rede. Esse particionamento faz com que os nós fiquem agrupados em **subredes** sem comunicação entre si. Veja:



Assim, somente os nós dentro da mesma subrede conseguem se comunicar para trocar mensagens, sincronizar, realizar o processamento distribuído das transações, etc. As subredes ficam sem conexão uma com a outra, impossibilitando o funcionamento do sistema em sua forma completa.

A propriedade de **tolerância a particionamento** determina precisamente que o sistema **deve continuar funcionando** nessas subredes enquanto o sistema não é retornado ao seu estado íntegro.

Fragmentação

Como mencionamos, a **fragmentação** de dados em um sistema distribuído diz respeito à técnica utilizada para dividir os dados das tabemas que compõem o modelo ao longo dos vários nós do sistema. A fragmentação pode ser **horizontal** ou **vertical**.

Na fragmentação **horizontal**, as tabelas sofrem "cortes" horizontais. Ou seja, parte das tuplas de cada relação fica em cada nó. Essa fragmentação costuma ser realizada com base no valor de um atributo. Por exemplo, uma empresa que possui uma filial em cada região do país, com um banco de dados parte do sistema distribuído em cada uma delas, poderá dividir os clientes com base no valor de seu atributo "Região". Assim, os clientes da região Norte ficarão no nó daquela região, enquanto que os clientes da região Sudeste ficariam no nó dessa outra, e assim sucessivamente.

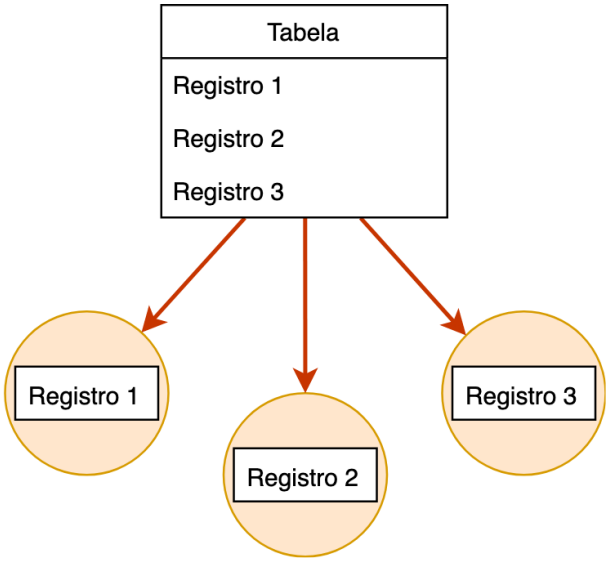


Figura: Ilustração de fragmentação horizontal

Como você pode ver, a repartição horizontal é feita se espalhando subconjuntos de tuplas da relação pelos nós. Nesse caso, temos que a fragmentação **mantém a mesma estrutura** da relação em cada um dos sítios, uma vez que todas as tuplas de uma relação têm a mesma estrutura. Para recompor a tabela original, pode-se utilizar uma operação de **união** entre as tuplas.

Elmasri & Navathe ressaltam que a **fragmentação horizontal derivada** aplica o particionamento de uma relação a outras relações secundárias relacionadas a ela. Por exemplo, se fragmentamos a tabela **departamento**, devemos fragmentar da mesma maneira outras relações que estão relacionadas a ela através de chave estrangeira, como **funcionário** e **projeto**.

A a fragmentação **vertical**, por sua vez, parte do pressuposto que os sítios podem ter necessidades diferentes em relação aos atributos da relação. Nesse caso, as relações são "cortadas" de forma vertical, sendo divididas em conjuntos de colunas.

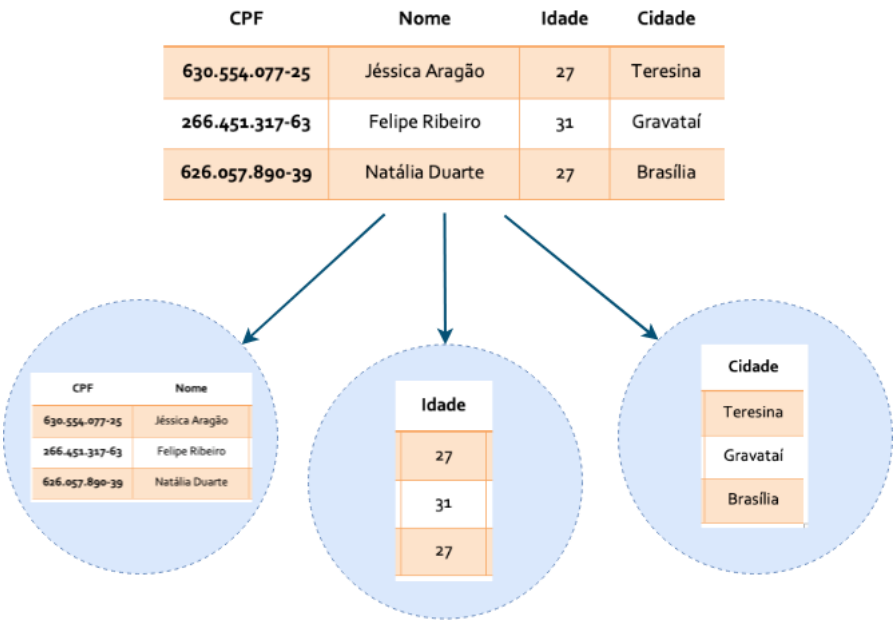


Figura: Ilustração de fragmentação vertical

Cada nó, então, ficará com um subconjunto das **colunas** ou **atributos** da relação. No entanto, isso pode trazer problemas se não tomarmos determinadas precauções. Se fizermos uma decomposição da tabela da mesma forma que fizemos no diagrama acima, não temos condições de juntá-la de volta quando estivermos realizando consultas. Observe que não há nenhuma informação que conecte uma idade ou cidade a uma pessoa específica.

Por esse motivo, Ramakrishnan (2002) aponta que essa fragmentação deve ser uma decomposição **sem perdas** *(lossless-join decomposition)*, devendo haver alguma maneira de unirmos novamente os fragmentos dos registros. Uma maneira de garantir que não haja perda de dados na fragmentação vertical é incluir um **identificador único das tuplas** em cada um dos fragmentos. Ou seja, no nosso exemplo, podemos incluir a **chave primária** da tabela (CPF) em todos os pedaços da tabela original:

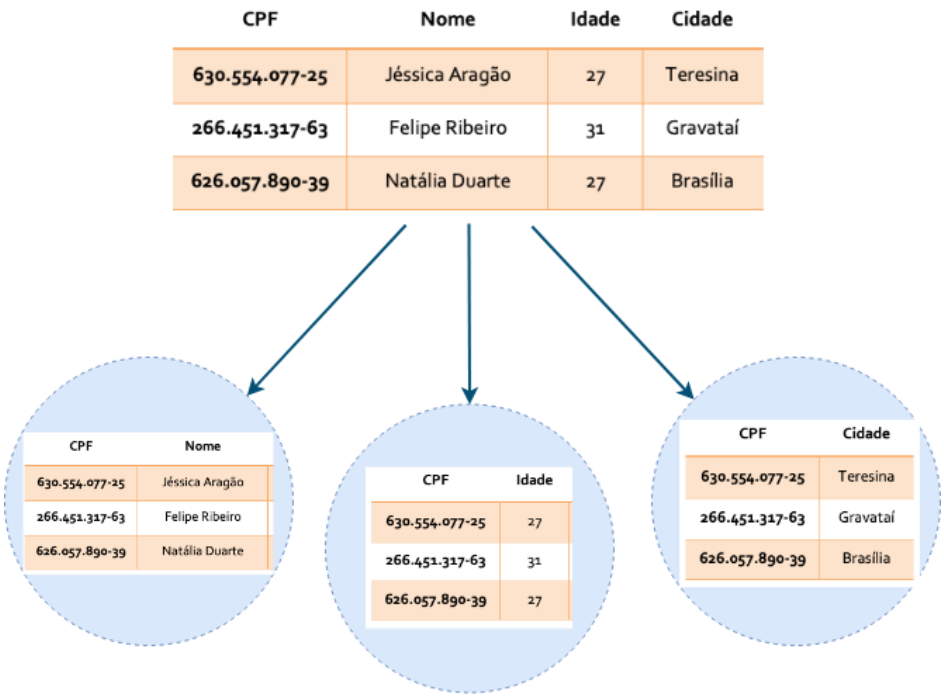


Figura: Ilustração de fragmentação vertical sem perdas

As fragmentações que mostramos até agora tem as características de serem **completas** e **disjuntas**. Completas porque a soma dos fragmentos inclui todos os elementos particionados - todas as linhas, no caso da horizontal, ou todas as colunas, no caso da vertical. Disjuntas pois os fragmentos **não têm uma interseção**, ou seja, na fragmentação horizontal, cada tupla está armazenada somente em um nó. Uma fragmentação vertical completamente disjunta não permitiria a recomposição sem perdas, já que não haveria atributo chave em comum para realizar a reconstrução do registro original através de junções.

Há também a possibilidade de se realizar uma relação **híbrida**, obtida através das operações de seleção e projeção da álgebra relacional (WHERE e SELECT na linguagem SQL). Os fragmentos resultantes, como você deve imaginar, serão compostos por subconjuntos tanto das **tuplas** como dos **atributos** da tabela original.

Replicação

Replicação é armazenar **diversas cópias** de uma relação ou de um fragmento de uma relação ao longo dos sítios. Ela pode ser **parcial** ou **completa**. Um banco de dados distribuído completamente replicado é aquele em que **todo dado x** está presente em **todos os sítios** que compõem o sistema. Já a replicação parcial é aquela em que nem todos os dados estão presentes em todos os sítios.

Na replicação **parcial**, podemos ter várias hipóteses. Os fragmentos podem estar replicados somente em alguns sítios, fragmentos distintos da mesma tabela podem ter regras de replicação diferentes (um fragmento sem replicação e outro fragmento com, por exemplo), e assim sucessivamente.

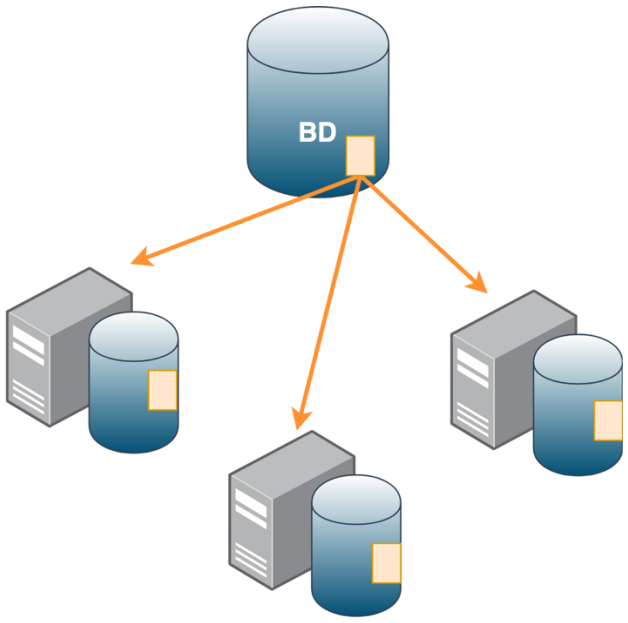


Figura: ilustração da replicação de um fragmento do banco de dados

A replicação pode ser realizada com dois objetivos (Ramakrishnan & Gehrke, 2002):

- **Aumentar a disponibilidade dos dados**, pois se um nó que contém a réplica dos dados se tornar indisponível, esses dados poderão ser encontrados em outros nós.
- **Melhorar a performance das consultas**, já que as consultas podem consultar uma cópia local dos dados no lugar de ter que esperar o retorno de um sítio remoto.

Além disso, Silberschatz (2010) aponta as seguintes características (vantajosas e desvantajosas) da replicação:

- **Maior disponibilidade**, como já vimos acima

- **Paralelismo aumentado**, já que os vários nós que contêm réplicas de uma determinada relação podem processar as consultas que a envolvem em paralelo
- **Aumento da sobrecarga nas atualizações**, uma vez que o sistema tem sempre que conferir a consistência das réplicas. Ou seja, quando se atualiza um determinado item em um dos nós, o **SGBDD** deve garantir que os demais nós que contêm réplicas do item também sejam atualizados. Assim, esse tipo de operação tende a ser mais custoso!

Replicação síncrona e assíncrona

Em relação ao **aspecto temporal**, a replicação pode ocorrer de maneira **síncrona** ou **assíncrona**.

A replicação **síncrona** determina que, antes de uma transação realizar o commit, ela deve **sincronizar** todas as cópias do dado que está sendo modificado, de modo a garantir que a alteração seja realizada em todos eles. No caso de um sistema distribuído em que os nós se encontram geograficamente muito distantes, com uma alta latência de comunicação de dados (ou seja, os dados **demoram** a serem transferidos entre os nós), a replicação síncrona não parece ser uma solução muito adequada, já que as transações irão demorar bastante para poderem ser realizadas, dada a sobrecarga de comunicação.

Já na replicação **assíncrona**, os dados são sincronizados de forma **periódica**, ou seja, em determinados momentos uma transação que lê múltiplas cópias do mesmo item de dados pode enxergar valores diferentes. Esse tipo de replicação não é adequado para uma aplicação distribuída em que a consistência imediata no valor dos dados seja essencial.

Por exemplo, se um banco utiliza um sistema distribuído com esse tipo de replicação, um cliente pode estar com um determinado saldo na conta em um nó e um valor diferente em outro nó. Isso obviamente traria problemas para o negócio!

Alocação

Nos bancos de dados distribuídos, cada fragmento de dados ou sua cópia, quando há replicação, deve ser **alocado** a um determinado sítio. Esse processo é chamado de **distribuição ou alocação de dados**. Quando todos os fragmentos são **disjuntos** (com a exceção da repetição das chaves primárias, que permitem a recomposição dos fragmentos verticais), a distribuição ou alocação é dita **não redundante** (Elmasri & Navathe, 2010).

Failover

Em um ambiente que possui replicação, a ação de **failover** é descrita como a operação de comutar ou trocar a operação de um nó principal para um nó alternativo. Por exemplo, se o nó responsável pela execução do serviço perde a conexão com a rede e fica off-line, outro nó assume o serviço e passa a receber e responder as requisições dos clientes.

No caso de falhas em um sistema de **alta disponibilidade**, essa transição para outro nó deve ocorrer de forma automática, sem a intervenção do administrador do sistema. Como o ambiente deve estar disponível no máximo de tempo possível, é importante que não haja atraso entre a ocorrência da falha e a volta do sistema ao ar. É

interessante que esse processo ocorra da forma mais **transparente** possível, de forma que o usuário não note que houve uma troca de nó.

Existe também o conceito de **switchover**, análogo ao failover, mas que geralmente é definido como algo mais **planejado**, de forma programada, com fins de administração do banco de dados. É uma troca de nó realizada geralmente de forma **manual** pelo administrador do banco. Também é conhecido como **failover manual planejado** em alguns sistemas, como o Microsoft SQL Server.

Para que não haja **perda de dados**, as réplicas deverão estar sincronizadas no momento do failover. A ação conhecida como failover forçado ou failover manual forçado consiste em substituir o nó que falhou por uma réplica não sincronizada. Esse tipo de failover provavelmente irá trazer uma perda de dados e só deve ser utilizado como última alternativa, como na recuperação de desastres (falhas catastróficas).

Processamento de transações

Para manter as propriedades **ACID** das transações, os sistemas distribuídos necessitam, além dos módulos de gerenciamento de transação locais de cada sítio e dos módulos de controle de concorrência e do gerenciador de recuperação de falhas do SGBD distribuído (**SGBDD**), de um **módulo global de gerenciamento de transações**.

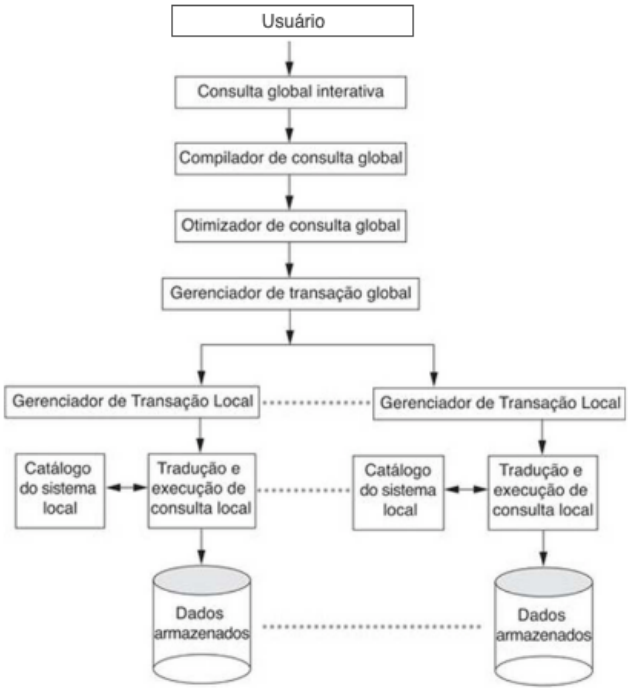


Figura: Arquitetura de um sistema de bancos de dados distribuído (Elmasri & Navathe, 2010)

Esse módulo tem como objetivo coordenar a execução das transações realizar as seguintes ações, de acordo com a operação de transação executada (Elmasri & Navathe, 2010):

READ: as operações de leitura devem retornar uma cópia local dos dados, quando disponível.

WRITE: as atualizações realizadas devem estar visíveis em todos os sítios em que o dado está replicado.

ABORT: os efeitos das transações abortadas não devem ser refletidos em nenhum dos sítios do sistema.

COMMIT: os efeitos das transações confirmadas devem ser refletidos em todos os nós que contêm cópias do item.

O protocolo Two-phase Commit

Outro conceito relativo às transações em bancos de dados distribuídos que cai em concursos é protocolo **two-phase commit (2PC)**. Esse protocolo serve para gerenciar as operações de ABORT/COMMIT das transações que são executadas em um SGBDD.

O gerenciamento de transações distribuídas tem um grau de dificuldade a mais em relação às transações convencionais, já que uma só transação pode envolver múltiplos bancos de dados. Assim, é necessário garantir as propriedades das transações, especialmente a atomicidade, em **todos os bancos de dados envolvidos**.

Para permitir esse gerenciamento, o protocolo **2PC** opera, como o nome já diz, em duas fases:

Fase 1:

- Todos os nós participantes devem avisar ao **coordenador**, o gerenciador global de recuperação, que a etapa da transação que envolve o nó foi concluída.
- Após receber o aviso de todos os nós que as partes da transação foram finalizadas, o coordenador manda uma mensagem para os nós participantes dizendo "*prepare-se para o commit" (prepare for commit)*.
- Ao receber a mensagem, os nós escrevem nos seus logs locais as informações que serão necessárias para reverter ou refazer as operações realizadas durante a transação, caso necessário.
    - Se o nó conseguiu escrever no log com sucesso e os efeitos da transação estão em condições de ser confirmados, ele responde *"pronto para o commit" ou OK (ready to commit/OK).*
    - Caso contrário, o nó responderá *"não posso realizar o commit" ou NÃO OK (cannot commit/not OK).* Se o coordenador não detectar resposta após um determinado período de tempo, ele irá assumir que a resposta foi pela impossibilidade de realizar o commit.

Fase 2:

- Se todos os nós responderam um **OK** e o coordenador também votou pelo **OK**, a transação é finalizada com **sucesso**.
    - Assim, o coordenador envia um sinal de *commit* para os nós participantes, que irão efetivar as transações localmente.
- No entanto, se algum nó ou o próprio coordenador responde um **NÃO OK**, a transação **falha**.
    - Nesse caso, o coordenador manda um sinal de *rollback*, indicando que os nós participantes devem reverter os efeitos das transações localmente.

No final das contas, podemos dizer que o **2PC** visa garantir que, ou todos os sítios podem efetivar a transação, ou nenhum deles o faz!

Esse algoritmo, no entanto, tem alguns problemas. Elmasri & Navathe indicam que o principal problema é que ele é um protocolo **bloqueante**. Isso leva às seguintes situações:

1) Uma falha no coordenador bloqueia todos os sítios até que o coordenador possa voltar, não deixando que eles modifiquem o recurso alvo da transação.
2) Quando ocorre uma falha simultânea entre o membro coordenador e um dos participantes na 2ª fase do protocolo. Pode acontecer o seguinte cenário:
   I. Todos os nós indicadaram que podem commitar *(ready to commit)*, juntamente com o coordenador.
   II. A segunda fase inicia, e o coordenador envia sinais de commit para todos os nós
   III. Nesse momento, tanto um nó participante quanto o coordenador **falham**, antes que esse nó participante possa escrever os dados no disco. Nesse cenário, os demais nós que já receberam o sinal de *commit* **confirmam** a transação, mas o nó que falhou a **reverte**, dada a ocorrência da falha. Da mesma maneira, como o coordenador também falhou, ele **não pode enxergar a falha** e avisar aos demais nós que a transação deve ser abortada.

Para resolver as limtações do **2PC**, foi criado o protocolo ***three-phase commit* (3PC)**, que separa a segunda fase em duas: **prepare-to-commit** (prepare-se para o commit) e **commit**. Essa fase de preparação serve para **informar** aos demais nós o resultado da votação. Assim, se o coordenador falhar, um outro nó pode tomar seu lugar e concluir a execução do protocolo da mesma maneira que seria feita no 2PC.

Atenção! Não confunda o protocolo ***two-phase commit*** com o ***two-phase locking***, que está relacionado aos bloqueios de controle de concorrência.

Teorema CAP

Vimos bastante sobre as propriedades de **disponibilidade**, **consistência** e **tolerância a particionamento** e como essas propriedades se relacionam com as diversas técnicas utilizadas nos sistemas distribuídos. Essas três propriedades são bastante **desejáveis** quando se trata de sistemas distribuídos.

Para que você se lembre, elas querem dizer o seguinte:

- **Consistência (*Consistency*)** – a consistência determina que os nós replicados devem ter representações consistentes dos dados, ou seja, as cópias de um mesmo dado devem ser idênticas.
- **Disponibilidade (*Availability*)** – o sistema deve estar disponível para processar transações de leitura ou escrita sobre os dados. Caso a operação não possa ser realizada com sucesso, o sistema deve retornar uma mensagem indicando a falha.
- **Tolerância a particionamento (*Partition Tolerance*)** – diz respeito à capacidade do sistema continuar operando no caso de uma falha de rede que ocasione particionamento, caso em que os nós são separados em subredes que não se comunicam uma com a outra.

Se você prestar atenção, vai ver que as iniciais das propriedades forma a sigla **CAP**. O teorema CAP é um princípio que determina que **não é possível garantir todas essas três propriedades ao mesmo tempo**. Os sistemas NoSQL, por exemplo, costumam aplicar uma **consistência eventual** para atingir as outras duas propriedades.

Como Elmasri & Navathe chamam a atenção, essa **consistência** dos nós replicados nada tem a ver com a consistência das propriedades **ACID** das transações. São conceitos diferentes, então analise bem o contexto das questões de concursos para não se confundir!

* * *

Por hoje é só! Terminamos o nosso conteúdo teórico. A seguir, uma bateria de questões e seus comentários para complementar seu conhecimento. Qualquer dúvida ou problema, fique à vontade para me contactar.

Bom proveito!

Questões comentadas pelo professor

1. (CESPE – PF – 2018)

No que se refere aos conceitos de estratégias de distribuição de banco de dados, julgue o item que se segue.

Disponibilidade de um sistema de banco de dados distribuído é, por definição, a característica de o sistema estar sempre disponível para ser utilizado imediatamente.

RESOLUÇÃO:

Tanto a disponibilidade quanto a confiabilidade são medidas relacionadas ao estado operacional do sistema distribuído. Se o sistema está acessível e funcionando corretamente quando o usuário precisa, ele é **confiável**. Se o sistema está sempre num estado acessível durante um período de tempo, ele é considerado com **alta disponibilidade**. Assim, o termo "sempre estar disponível" nos leva a crer que a assertiva está correta.

Gabarito: C

2. (CESPE – PF – 2018)

No que se refere aos conceitos de estratégias de distribuição de banco de dados, julgue o item que se segue.

Na replicação síncrona, recomenda-se que os bancos de dados fiquem armazenados em sítios geograficamente distantes entre si, pois a execução da replicação ocorrerá com um atraso, que varia de poucos minutos a horas.

RESOLUÇÃO:

Nesse caso, a replicação síncrona não parece ser a melhor opção, já que o atraso irá inviabilizar a execução tempestiva das transações, uma vez que os nós precisarão se comunicar e sincronizar a cada modificação realizada em uma das cópias. A replicação **assíncrona**, técnica em que há uma sincronização periódica dos nós, é a solução mais adequada para esse cenário.

Gabarito: E

3. (CESPE – EBSERH – 2018)

Julgue o item seguinte, a respeito de banco de dados distribuído e orientado a objetos.

Em um ambiente distribuído, as diferentes topologias de redes utilizadas para a comunicação entre os bancos de dados não interferem no desempenho dos bancos de dados quanto ao processamento de consultas, uma vez que as consultas são executadas diretamente nos servidores.

RESOLUÇÃO:

As diferentes formas de organização de rede (topologias), a distância entre os nós, as velocidades dos links de conexão, a latência, tudo isso irá interferir sim no desempenho dos bancos de dados distribuídos. Como esse tipo de sistema opera de forma **coordenada** com os outros, mesmo que uma transação seja iniciada em um nó, é necessária a comunicação com os demais nós para garantir, por exemplo, que não há valores distintos para as diferentes cópias de um determinado objeto.

Gabarito: E

4. (CESPE – EBSERH – 2018)

Julgue o item seguinte, a respeito de banco de dados distribuído e orientado a objetos.

Em um banco de dados distribuído, os servidores de banco envolvidos não precisam, necessariamente, possuir a mesma configuração de hardware.

RESOLUÇÃO:

Uma das características distintivas dos bancos de dados distribuídos é justamente a **ausência de restrição de homogeneidade** em relação aos nós conectados ao sistema. Assim, cada nó pode ter sua própria configuração de hardware e software ou mesmo seu próprio conjunto de dados.

Gabarito: C

5. (CESPE – ABIN – 2018)

Acerca de ambientes de alta disponibilidade e escalabilidade, *fail-over* e técnicas de detecção de problemas e otimização de desempenho, julgue o item que se segue.

Em um sistema gerenciador de correio eletrônico, o processo de *fail-over* não necessariamente deve ser totalmente transparente e automático, pois pode haver a intervenção de um administrador do sistema, principalmente se houver a necessidade de reconexão manual do cliente da aplicação.

RESOLUÇÃO:

Veja que o enunciado trata de um sistema de **alta disponibilidade**. Nesse caso, é importante que o processo de failover se dê de forma **automática**, sem a necessidade da reconexão manual da aplicação por parte do administrador do sistema.

Gabarito: E

6. (CESPE – ABIN – 2018)

Acerca de ambientes de alta disponibilidade e escalabilidade, fail-over e técnicas de detecção de problemas e otimização de desempenho, julgue o item que se segue.

É chamado de escalável um sistema gerenciador de correio web que mantenha o mesmo desempenho se a capacidade da infraestrutura aumentar na mesma proporção que o tamanho do problema.

RESOLUÇÃO:

A questão traz uma definição interessante de escalabilidade. Essa propriedade diz respeito à capacidade do sistema se expandir para lidar com o aumento dos requisitos de carga de trabalho. Se o problema aumenta e a infraestrutura pode ser expandida de acordo, sem que o sistema falhe, podemos dizer que este é escalável.

Gabarito: C

7. (CESPE – ABIN – 2018)

Acerca de sistemas de aplicação web, julgue o item a seguir.

Replicação é o processo de criação de uma cópia exata (réplica) do dado, a qual será usada para restaurar as operações, caso haja perda de dados. Em servidores de aplicação, a replicação local não necessariamente ocorre no mesmo datacenter ou array.

RESOLUÇÃO:

A replicação **local** indica que as réplicas estão dentro da mesma estrutura, ou seja, no mesmo datacenter ou array de servidores. O contrário de replicação local seria replicação **remota**, em algum sítio fisicamente distante do principal.

Gabarito: E

8. (CESPE – SEDF – 2017)

Julgue o próximo item, relativos à tecnologia de bancos de dados distribuídos.

Uma relação ou uma tabela pode estar fragmentada e armazenada em pontos diferentes; nesse caso, quando se separam os registros (linhas) da tabela, tem-se uma fragmentação horizontal.

RESOLUÇÃO:

A fragmentação horizontal determina que são realizados cortes **horizontais** na relação, separando-a em vários subconjuntos de suas tuplas ou registros. Assim, a alternativa está correta!

Gabarito: C

9. (CESPE – SEDF – 2017)

Julgue o próximo item, relativos à tecnologia de bancos de dados distribuídos.

Uma desvantagem dos bancos de dados distribuídos é a falta de autonomia local, visto que um banco X depende da sincronização com um banco Y para que as operações sejam bem-sucedidas.

RESOLUÇÃO:

Os bancos de dados distribuídos não necessariamente serão **completamente distribuídos**, sem nenhum grau de autonomia para seus participantes. Os sistemas **federados**, por exemplo, mantêm a autonomia dos nós, com a condição que haja um **esquema ou visão global**.

Gabarito: E

10. (CESPE – TCE/PA – 2016)

No que se refere à tipologia de ambientes com alta disponibilidade e escalabilidade para a estruturação de ambientes computacionais, julgue o item subsequente.

Denomina-se *failover* o processo, transparente ou não, em que um nó assume o funcionamento de outro nó em razão de este ter apresentado alguma falha.

RESOLUÇÃO:

Principalmente no contexto de ambientes de alta disponibilidade, assume-se que failover é o processo da comutação, ou seja, da substituição de um nó por outro de forma **automática**. Isso ocorre no caso de falha com o nó principal e deve ocorrer, preferencialmente, de forma transparente, ou seja, invisível para o usuário.

Na questão, vemos que a expressão "transparente ou não" não invalidou o gabarito correto, então podemos assumir que o CESPE adota o entendimento que o failover pode sim ocorrer de forma não transparente, apesar de não ser a melhor opção!

Gabarito: C

11. (CESPE – TCU – 2015 – ADAPTADA)

Considere as seguintes configurações de servidores de banco de dados (SBD):

...

III- Dois servidores foram configurados para trabalhar de forma distribuída do tipo SBDF (sistemas de banco de dados federado), com intuito primordial de garantir mais disponibilidade, no caso de falha de um dos servidores.

...

Com base nessas configurações, julgue o item abaixo.

A configuração III, além de mais disponibilidade, provê, também, mais confiabilidade, pois há isolamento de falhas, ou seja, uma falha que afete um dos servidores não afeta o outro. Além disso, é certo que existe alguma visão ou esquema global da federação de banco de dados compartilhada pelas aplicações.

RESOLUÇÃO:

A assertiva pode ser dividida em duas partes:

"A configuração III, além de mais disponibilidade, provê, também, mais confiabilidade, pois há isolamento de falhas, ou seja, uma falha que afete um dos servidores não afeta o outro." – Essa parte está correta, já que os sistemas distribuídos incluem técnicas de replicação. Assim, caso um nó falhe, será possível acessar algum dos outros nós disponíveis. Isso aumenta a confiabilidade, já que aumenta a probabilidade de uma requisição por parte dos usuários em um determinado ponto no tempo ser atendida com sucesso.

"Além disso, é certo que existe alguma visão ou esquema global da federação de banco de dados compartilhada pelas aplicações." – A segunda parte também está correta. Os sistemas federados contam com um grande nível de autonomia local, mas há sempre a presença de um **esquema ou visão global**.

Gabarito: C

12. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

A replicação total implica cópia das tabelas em todos os servidores envolvidos no projeto, de modo a aumentar a disponibilidade dos dados para que o sistema continue a processar consultas que envolvam as tabelas independentemente da falha de algum servidor.

RESOLUÇÃO:

A replicação total implica que uma determinada relação estará replicada em **todos os nós participantes** do sistema distribuído. Esse tipo de organização implica uma maior redundância e maior disponibilidade, ao tornar mais provável que um determinado de dado fique disponível, mesmo em caso de falha em algum dos nós.

Gabarito: C

---

13. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Em um sistema gerenciador de banco de dados distribuídos, o acesso transparente ocorre quando usuários interagem com o sistema como se este fosse um único sistema lógico.

RESOLUÇÃO:

O objetivo do conceito de transparência nos bancos de dados distribuídos é justamente tornar o processo de distribuição invisível ao usuário. Assim, ele terá a mesma visão dos dados que teria caso estivesse usando um sistema centralizado convencional, sem precisar se importar com os detalhes da distribuição.

Gabarito: C

---

14. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Um banco de dados distribuído é aquele cujos dados estão armazenados em diversos bancos de dados localizados em sítios distintos, o que gera como resultado a união lógica desses bancos de dados reais.

RESOLUÇÃO:

Perfeito! Um dos pressupostos dos bancos de dados distribuídos é justamente a existência de uma interconexão lógica entre os bancos de dados em sítios distintos. Mesmo que esses sítios estejam em localidades fisicamente distantes, a união lógica irá permitir a efetivação da distribuição.

Gabarito: C

---

15. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Em um ambiente distribuído de banco de dados, a confiabilidade consiste em uma coleção de sistemas de monitoramento da atividade de tráfego de informações sobre a rede de interconexão dos servidores de aplicação.

RESOLUÇÃO:

A confiabilidade não é uma coleção de sistemas, é uma **propriedade** que diz respeito à capacidade do sistema de estar disponível e funcionando corretamente em um determinado ponto do tempo, quando receber alguma requisição do usuário.

Gabarito: E

---

16. (CESPE – SUFRAMA – 2014)

Julgue os itens subsequentes, relativos a bancos de dados.

Um sistema de gerenciamento de bancos de dados distribuídos (SGBDD) é o resultado da fusão das tecnologias de banco de dados e de redes e comunicação de dados. Nesse sistema, os elementos de processamento são necessariamente homogêneos e permitem que o processamento e a entrega de dados seja centralizada ou distribuída, diferentemente dos SGBDs tradicionais, que realizam esse processamento exclusivamente na forma centralizada.

RESOLUÇÃO:

Os SGBDs distribuídos realmente utilizam conceitos de bancos de dados de de redes de computadores. No entanto, os nós **não precisam ser homogêneos**, sendo essa uma das principais características desse tipo de sistema. Nesse tipo de SGBD, o processamento e a entrega dos dados realmente podem ocorrer de forma local, a depender do grau de autonomia dos nós.

Gabarito: E

---

17. (CESPE – Banco da Amazônia – 2012)

Acerca de conceitos de banco de dados, características dos bancos relacionais e linguagem SQL, julgue os itens que se seguem.

Durante um failover, existindo replicação do banco de dados, não haverá perda de dados, ainda que o administrador de banco de dados intervenha manualmente para o retorno do funcionamento do banco de dados principal.

RESOLUÇÃO:

Um failover manual **forçado** pode acontecer em uma situação em que a réplica que irá substituir a principal não está perfeitamente sincronizada. Nesse caso, poderá haver sim perda de dados.

Gabarito: E

18. (CESPE – Banco da Amazônia – 2012)

Acerca de segurança da informação, julgue os itens subsecutivos.

Failover é a capacidade que um sistema possui de continuar, automaticamente, um serviço em caso de falhas, sem a necessidade de um comando humano.

RESOLUÇÃO:

Quando a questão não especificar o tipo do failover, assuma que se trata do **failover automático**, o padrão, que ocorre em caso de falhas de forma automática e transparente. Nesse caso, não há a necessidade da intervenção humana para a substituição do nó que falhou por uma réplica disponível. Item correto!

Gabarito: C

19. (FCC – DPE/AM – 2018)

Considerando a replicação de tabelas em bancos de dados relacionais, se a maioria dos acessos for apenas de leitura de uma tabela T, vários locais podem processar consultas de forma paralela, envolvendo acesso a essa tabela T. Tal característica é denominada

a) sobrecarga atenuada.

b) transparência parcial.

c) transação em duas fases.

d) transparência de replicação.

e) paralelismo aumentado.

RESOLUÇÃO:

A característica positiva da replicação que diz respeito à possibilidade da realização de consultas em paralelo nas diferentes cópias de uma relação é justamente o **paralelismo aumentado**. Com a utilização de paralelismo, as transações podem executar de forma mais célere, já que as tarefas podem ser divididas entre as diferentes cópias da tabela e executadas de forma simultânea.

Gabarito: E

20. (FCC – DPE/AM – 2018)

Em bancos de dados relacionais normalmente há uma política de replicação de dados. Sobre essa política,

a) o número de cópias ou réplicas de um conjunto de dados deve ser impar, seja no caso síncrono, seja no assíncrono.

b) na replicação do tipo síncrono, as réplicas são atualizadas em uma nova transação realizada posteriormente à transação originária de uma modificação, e portanto nem todas as réplicas terão valores idênticos dos dados.

c) a replicação representa uma melhor disponibilidade do banco de dados, pelo fato de haver réplica(s) disponível(eis) na eventualidade da ocorrência de falhas no sistema.

d) na replicação do tipo assíncrono, as réplicas devem ser atualizadas na mesma transação originária de uma modificação, resultando em apenas uma versão dos dados.

e) na chamada independência ou transparência de replicação, os usuários devem conhecer o número e a localização das réplicas.

RESOLUÇÃO:

a) Não existe restrição de número par ou ímpar de cópias, seja na replicação síncrona ou assíncrona. **ERRADA**

b) Na replicação síncrona, as atualizações são escritas em todas as cópias **antes** da confirmação da transação. Isso garante que não haverá representações distintas para o mesmo item de dados no sistema distribuído. **ERRADA**

c) Uma das características positivas da replicação é precisamente o **aumento da disponibilidade**. Como há várias cópias dos dados, mesmo que parte delas fique indisponível, ainda haverá outras instâncias do mesmo item de dados que podem ser utilizadas nas transações. **CERTA**

d) Na replicação assíncrona, as réplicas são atualizadas periodicamente, em outra transação que não é a originária da modificação. Isso pode ocasionar variações nas versões dos dados. **ERRADA**

e) Na transparência de replicação, os usuários não precisam tomar conhecimento de que uma tabela está replicada ou das características dessa replicação. **ERRADA**

Gabarito: C

21. (FCC – DPE/RS – 2017)

Em sistemas de bancos de dados distribuídos utiliza-se, frequentemente, a técnica de replicação de tabelas, sendo que tal técnica apresenta como características proporcionar

a) menor disponibilidade, maior problema de concorrência e menor sobrecarga na atualização de réplicas.

b) menor disponibilidade, maior problema de concorrência e maior sobrecarga na atualização de réplicas.

c) maior disponibilidade, maior problema de concorrência e menor sobrecarga na atualização de réplicas.

d) menor disponibilidade, menor problema de concorrência e menor sobrecarga na atualização de réplicas.

e) maior disponibilidade, menor problema de concorrência e maior sobrecarga na atualização de réplicas.

RESOLUÇÃO:

Os bancos de dados distribuídos apresentam **maior disponibilidade**, já que, se um nó fica indisponível, as transações que desejam acessar determinado dado podem se dirigir aos demais nós disponíveis. No entanto, há uma **maior sobrecarga** na ocorrência de atualizações de dados, já que as réplicas devem sincronizar para garantir que as várias cópias contêm o mesmo valor para um mesmo item de dados.

Com isso, já podemos responder que a alternativa correta é a letra E.

Em relação ao **menor problema de concorrência**, isso se dá porque temos vários pontos locais de acesso aos dados, não somente um. Assim, em teoria, haveira menor ocorrência de conflitos. No entanto, o gerenciamento de concorrência é bem mais complexo, já que deve se garantir que as várias transações distribuídas ocorram de forma consistente em relação a todos os nós. Considero essa parte da alternativa um pouco dúbia, mas essa é a única alternativa plausível.

Gabarito: E

22. (FCC – DPE/SP – 2015)

Para realizar atualização de dados em bancos de dados distribuídos pode ser utilizada uma estratégia onde as cópias de uma relação modificada são atualizadas apenas periodicamente e uma transação que leia cópias diferentes da mesma relação poderá ver valores diferentes por um tempo, podendo comprometer, nesse caso, a independência dos dados distribuídos. Esta estratégia é conhecida como:

a) replicação síncrona.

b) fragmentação horizontal.

c) fragmentação consistente.

d) replicação assíncrona.

e) fragmentação vertical.

RESOLUÇÃO:

A modalidade de replicação que não garante a sincronia dos dados é a replicação **assíncrona**. Esse modo de recuperação é adequado para aqueles ambientes em que há uma grande latência de comunicação entre os sítios ou que têm uma necessidade da diminuição da sobrecarga nas atualizações, por exemplo.

Gabarito: D

23. (FGV – Pref. de Niterói – 2018)

Uma grande dificuldade na implementação de bancos de dados que suportam transações distribuídas é a possibilidade de que uma transação seja apenas parcialmente concluída, de forma a criar inconsistências nas bases de dados. Isso pode ocorrer, por exemplo, por falhas de comunicação entre os nós envolvidos na transação.

Assinale a opção que indica o algoritmo usualmente empregado na execução de transações distribuídas, visando à manutenção das suas propriedades.

a) Armstrong's algorithm.

b) Begin-end algorithm.

c) Three-fase write.

d) Two-phase commit.

e) Two-phase lock.

RESOLUÇÃO:

O algoritmo que lida com as operações de **ABORT/COMMIT** das transações, com vistas à manter suas propriedades **ACID** é o *two-phase commit*. Esse algoritmo funciona em duas fases, tendo o objetivo de verificar que todos os nós podem realizar o commit - caso isso não aconteça, a transação é abortada em todos eles.

Gabarito: D

24. (FGV – IBGE – 2017)

Em relação à gerencia de transações em bancos de dados distribuídos, analise as afirmativas abaixo:

I. O protocolo three-phase commit visa solucionar uma falha do protocolo two-phase commit, quando ocorre falha simultânea do membro coordenador e de algum dos participantes na 2ª fase desse último protocolo.

II. No protocolo three-phase commit, os participantes têm igual hierarquia, não há mais a figura do coordenador que existe no protocolo two-phase commit.

III. No protocolo two-phase commit, existe uma fase inicial de votação, onde o coordenador envia uma solicitação de commit para todos os participantes, e depois cada um envia sua resposta (concordando ou não com o commit) para todos os demais.

Está correto somente o que se afirma em:

a) I;

b) II;

c) III;

d) I e II;

e) I e III.

RESOLUÇÃO:

I. O protocolo 3PC visa solucionar os problemas ocasionados pelo fato do 2PC ser bloqueante, em caso de falhas no coordenador. Um dos problemas do 2PC é exatamente o que ocorre quando tanto o coordenador quanto um nó falham na segunda fase, o que pode ocasionar inconsistências de dados. **CERTA**

II. No 3PC existe o coordenador. A diferença é que os nós participantes são informados do resultado da votação, permitindo que um deles assuma o papel de coordenador em caso de falha deste. **ERRADA**

III. No 2PC, os participantes respondem se podem ou não realizar o commit direto para o **coordenador**, não para os demais nós. **ERRADA**

Gabarito: A

Lista de questões comentadas

1. (CESPE – PF – 2018)

No que se refere aos conceitos de estratégias de distribuição de banco de dados, julgue o item que se segue.

Disponibilidade de um sistema de banco de dados distribuído é, por definição, a característica de o sistema estar sempre disponível para ser utilizado imediatamente.

2. (CESPE – PF – 2018)

No que se refere aos conceitos de estratégias de distribuição de banco de dados, julgue o item que se segue.

Na replicação síncrona, recomenda-se que os bancos de dados fiquem armazenados em sítios geograficamente distantes entre si, pois a execução da replicação ocorrerá com um atraso, que varia de poucos minutos a horas.

3. (CESPE – EBSERH – 2018)

Julgue o item seguinte, a respeito de banco de dados distribuído e orientado a objetos.

Em um ambiente distribuído, as diferentes topologias de redes utilizadas para a comunicação entre os bancos de dados não interferem no desempenho dos bancos de dados quanto ao processamento de consultas, uma vez que as consultas são executadas diretamente nos servidores.

4. (CESPE – EBSERH – 2018)

Julgue o item seguinte, a respeito de banco de dados distribuído e orientado a objetos.

Em um banco de dados distribuído, os servidores de banco envolvidos não precisam, necessariamente, possuir a mesma configuração de hardware.

5. (CESPE – ABIN – 2018)

Acerca de ambientes de alta disponibilidade e escalabilidade, *fail-over* e técnicas de detecção de problemas e otimização de desempenho, julgue o item que se segue.

Em um sistema gerenciador de correio eletrônico, o processo de *fail-over* não necessariamente deve ser totalmente transparente e automático, pois pode haver a intervenção de um administrador do sistema, principalmente se houver a necessidade de reconexão manual do cliente da aplicação.

6. (CESPE – ABIN – 2018)

Acerca de ambientes de alta disponibilidade e escalabilidade, fail-over e técnicas de detecção de problemas e otimização de desempenho, julgue o item que se segue.

É chamado de escalável um sistema gerenciador de correio web que mantenha o mesmo desempenho se a capacidade da infraestrutura aumentar na mesma proporção que o tamanho do problema.

7. (CESPE – ABIN – 2018)

Acerca de sistemas de aplicação web, julgue o item a seguir.

Replicação é o processo de criação de uma cópia exata (réplica) do dado, a qual será usada para restaurar as operações, caso haja perda de dados. Em servidores de aplicação, a replicação local não necessariamente ocorre no mesmo datacenter ou array.

8. (CESPE – SEDF – 2017)

Julgue o próximo item, relativos à tecnologia de bancos de dados distribuídos.

Uma relação ou uma tabela pode estar fragmentada e armazenada em pontos diferentes; nesse caso, quando se separam os registros (linhas) da tabela, tem-se uma fragmentação horizontal.

9. (CESPE – SEDF – 2017)

Julgue o próximo item, relativos à tecnologia de bancos de dados distribuídos.

Uma desvantagem dos bancos de dados distribuídos é a falta de autonomia local, visto que um banco X depende da sincronização com um banco Y para que as operações sejam bem-sucedidas.

10. (CESPE – TCE/PA – 2016)

No que se refere à tipologia de ambientes com alta disponibilidade e escalabilidade para a estruturação de ambientes computacionais, julgue o item subsequente.

Denomina-se *failover* o processo, transparente ou não, em que um nó assume o funcionamento de outro nó em razão de este ter apresentado alguma falha.

11. (CESPE – TCU – 2015 – ADAPTADA)

Considere as seguintes configurações de servidores de banco de dados (SBD):

...

III- Dois servidores foram configurados para trabalhar de forma distribuída do tipo SBDF (sistemas de banco de dados federado), com intuito primordial de garantir mais disponibilidade, no caso de falha de um dos servidores.

...

Com base nessas configurações, julgue o item abaixo.

A configuração III, além de mais disponibilidade, provê, também, mais confiabilidade, pois há isolamento de falhas, ou seja, uma falha que afete um dos servidores não afeta o outro. Além disso, é certo que existe alguma visão ou esquema global da federação de banco de dados compartilhada pelas aplicações.

12. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

A replicação total implica cópia das tabelas em todos os servidores envolvidos no projeto, de modo a aumentar a disponibilidade dos dados para que o sistema continue a processar consultas que envolvam as tabelas independentemente da falha de algum servidor.

13. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Em um sistema gerenciador de banco de dados distribuídos, o acesso transparente ocorre quando usuários interagem com o sistema como se este fosse um único sistema lógico.

14. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Um banco de dados distribuído é aquele cujos dados estão armazenados em diversos bancos de dados localizados em sítios distintos, o que gera como resultado a união lógica desses bancos de dados reais.

15. (CESPE – MEC – 2015)

No que diz respeito a bancos de dados distribuídos, julgue o próximo item.

Em um ambiente distribuído de banco de dados, a confiabilidade consiste em uma coleção de sistemas de monitoramento da atividade de tráfego de informações sobre a rede de interconexão dos servidores de aplicação.

16. (CESPE – SUFRAMA – 2014)

Julgue os itens subsequentes, relativos a bancos de dados.

Um sistema de gerenciamento de bancos de dados distribuídos (SGBDD) é o resultado da fusão das tecnologias de banco de dados e de redes e comunicação de dados. Nesse sistema, os elementos de processamento são

necessariamente homogêneos e permitem que o processamento e a entrega de dados seja centralizada ou distribuída, diferentemente dos SGBDs tradicionais, que realizam esse processamento exclusivamente na forma centralizada.

17. (CESPE – Banco da Amazônia – 2012)

Acerca de conceitos de banco de dados, características dos bancos relacionais e linguagem SQL, julgue os itens que se seguem.

Durante um failover, existindo replicação do banco de dados, não haverá perda de dados, ainda que o administrador de banco de dados intervenha manualmente para o retorno do funcionamento do banco de dados principal.

18. (CESPE – Banco da Amazônia – 2012)

Acerca de segurança da informação, julgue os itens subsecutivos.

Failover é a capacidade que um sistema possui de continuar, automaticamente, um serviço em caso de falhas, sem a necessidade de um comando humano.

19. (FCC – DPE/AM – 2018)

Considerando a replicação de tabelas em bancos de dados relacionais, se a maioria dos acessos for apenas de leitura de uma tabela T, vários locais podem processar consultas de forma paralela, envolvendo acesso a essa tabela T. Tal característica é denominada

a) sobrecarga atenuada.

b) transparência parcial.

c) transação em duas fases.

d) transparência de replicação.

e) paralelismo aumentado.

20. (FCC – DPE/AM – 2018)

Em bancos de dados relacionais normalmente há uma política de replicação de dados. Sobre essa política,

a) o número de cópias ou réplicas de um conjunto de dados deve ser impar, seja no caso síncrono, seja no assíncrono.

b) na replicação do tipo síncrono, as réplicas são atualizadas em uma nova transação realizada posteriormente à transação originária de uma modificação, e portanto nem todas as réplicas terão valores idênticos dos dados.

c) a replicação representa uma melhor disponibilidade do banco de dados, pelo fato de haver réplica(s) disponível(eis) na eventualidade da ocorrência de falhas no sistema.

d) na replicação do tipo assíncrono, as réplicas devem ser atualizadas na mesma transação originária de uma modificação, resultando em apenas uma versão dos dados.

e) na chamada independência ou transparência de replicação, os usuários devem conhecer o número e a localização das réplicas.

21. (FCC – DPE/RS – 2017)

Em sistemas de bancos de dados distribuídos utiliza-se, frequentemente, a técnica de replicação de tabelas, sendo que tal técnica apresenta como características proporcionar

a) menor disponibilidade, maior problema de concorrência e menor sobrecarga na atualização de réplicas.

b) menor disponibilidade, maior problema de concorrência e maior sobrecarga na atualização de réplicas.

c) maior disponibilidade, maior problema de concorrência e menor sobrecarga na atualização de réplicas.

d) menor disponibilidade, menor problema de concorrência e menor sobrecarga na atualização de réplicas.

e) maior disponibilidade, menor problema de concorrência e maior sobrecarga na atualização de réplicas.

22. (FCC – DPE/SP – 2015)

Para realizar atualização de dados em bancos de dados distribuídos pode ser utilizada uma estratégia onde as cópias de uma relação modificada são atualizadas apenas periodicamente e uma transação que leia cópias diferentes da mesma relação poderá ver valores diferentes por um tempo, podendo comprometer, nesse caso, a independência dos dados distribuídos. Esta estratégia é conhecida como:

a) replicação síncrona.

b) fragmentação horizontal.

c) fragmentação consistente.

d) replicação assíncrona.

e) fragmentação vertical.

23. (FGV – Pref. de Niterói – 2018)

Uma grande dificuldade na implementação de bancos de dados que suportam transações distribuídas é a possibilidade de que uma transação seja apenas parcialmente concluída, de forma a criar inconsistências nas bases de dados. Isso pode ocorrer, por exemplo, por falhas de comunicação entre os nós envolvidos na transação.

Assinale a opção que indica o algoritmo usualmente empregado na execução de transações distribuídas, visando à manutenção das suas propriedades.

a) Armstrong's algorithm.

b) Begin-end algorithm.

c) Three-fase write.

d) Two-phase commit.

e) Two-phase lock.

24. (FGV – IBGE – 2017)

Em relação à gerencia de transações em bancos de dados distribuídos, analise as afirmativas abaixo:

I. O protocolo three-phase commit visa solucionar uma falha do protocolo two-phase commit, quando ocorre falha simultânea do membro coordenador e de algum dos participantes na 2ª fase desse último protocolo.

II. No protocolo three-phase commit, os participantes têm igual hierarquia, não há mais a figura do coordenador que existe no protocolo two-phase commit.

III. No protocolo two-phase commit, existe uma fase inicial de votação, onde o coordenador envia uma solicitação de commit para todos os participantes, e depois cada um envia sua resposta (concordando ou não com o commit) para todos os demais.

Está correto somente o que se afirma em:

a) I;

b) II;

c) III;

d) I e II;

e) I e III.

Gabarito

1. C	9. E	17. E
2. E	10. C	18. C
3. E	11. C	19. E
4. C	12. C	20. C
5. E	13. C	21. E
6. C	14. C	22. D
7. E	15. E	23. D
8. C	16. E	24. A

Resumo direcionado

BDs distribuídos

Características

- São compostos por máquinas individuais que realizam cooperação na execução de tarefas

- **Conexão dos nós de banco de dados através de uma rede de computadores**
- **Interrelação lógica entre os bancos de dados conectados**
- **Ausência de necessidade de homogeneidade dos nós**

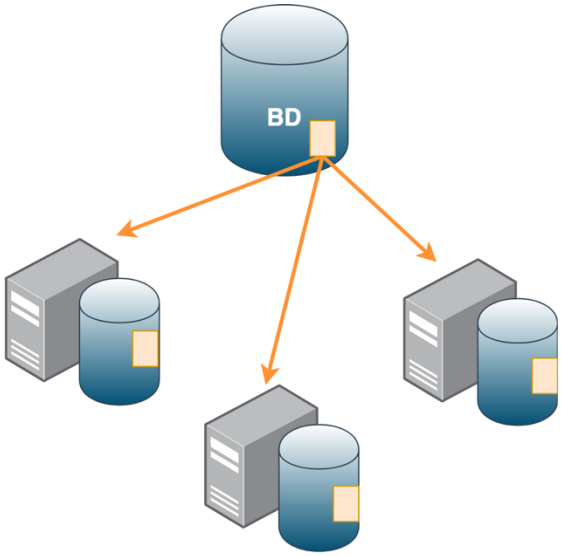
Propriedades

- **Confiabilidade** – sistema deve estar acessível e funcionar corretamente quando solicitado
- **Disponibilidade** – medida da capacidade do sistema de estar acessível ao longo do tempo, tem relação com o conceito de **continuidade**

- **Transparência** – as características dos sistemas distribuídos podem ser omitidas para o usuário, que deve enxergar um só sistema lógico:
  - **Transparência de organização de dados**
    - De **localização** ou de **nomeação**
  - **Transparência de replicação**
  - **Transparência de fragmentação**
  - **Outras formas de transparência, como transparência de design e de execução**

- **Autonomia** – capacidade dos nós operarem de forma individual
  - **Sistemas federados** possuem alto grau de autonomia, mas deve haver um **esquema global**

- **Fragmentação**
  - Relações são "partidas" e distribuídas ao longo dos nós
  - Pode ser **horizontal**: subconjuntos de tuplas são distribuídos/alocados ao longo dos nós
  - Ou **vertical**: conjuntos de atributos são separados, mas deve haver um identificador único da tupla para permitir a **recomposição** da tupla original, sem perdas!



- **Replicação**
  - Armazenamento de cópias dos dados nos diversos sítios
  - Pode ser total ou parcial
  - **Objetivos:**
    - Aumentar a disponibilidade
    - Melhorar performance
  - **Características:**
    - **Maior disponibilidade**
    - **Paralelismo aumentado**
    - **Aumento da sobrecarga nas atualizações**

- A replicação pode ser **síncrona** ou **assíncrona**
  - Síncrona: ocorre uma sincronização dos nós antes da persistência dos efeitos da transação
  - Assíncrona: dados são sincronizados de forma periódica, ou seja, pode haver inconsistências temporárias

Transações distribuídas

- As transações nos SGBDDs podem ocorrer em vários bancos de dados ao mesmo tempo
- Assim, devem ser garantidas determinadas características:

READ: as operações de leitura devem retornar uma cópia local dos dados, quando disponível.

WRITE: as atualizações realizadas devem estar visíveis em todos os sítios em que o dado está replicado.

ABORT: os efeitos das transações abortadas não devem ser refletidos em nenhum dos sítios do sistema.

COMMIT: os efeitos das transações confirmadas devem ser refletidos em todos os nós que contêm cópias do item.

- O protocolo **2PC** tem duas fases, uma de votação e outra em que há o commit ou o rollback dos efeitos
  - O propósito é garantir que, se um nó não pode realizar o commit, ninguém irá realizar!
  - Esse protocolo, no entanto, é bloqueante e apresenta falhas caso o coordenador falhe junto com um nó, na segunda fase
  - O **3PC** oferece uma melhoria ao permitir que um nó participante assuma a transação em caso de falha do coordenador

Failover

- Técnica utilizada para a substituição de um nó que falhou por alguma réplica disponível
- Costuma ocorrer de forma **automática**, sem a necessidade de intervenção do administrador
- É desejável que ocorra de forma **transparente** e sem perda de dados

Bibliografia

Elmasri, R., & Navathe, S. B. (2010). *Sistemas de Bancos de Dados* (6ª Ed.). São Paulo: Pearson Universidades.

Meyer, A. (2017). Types of backup: Full, Incremental, Differential, and Others. Retrieved August 16, 2019, from Nakivo Blog website: https://www.nakivo.com/blog/backup-types-explained-full-incremental-differential-synthetic-and-forever-incremental/

Ramakrishnan, R., & Gehrke, J. (2002). *Database Management Systems* (3rd Ed.). McGraw-Hill Education.

Sheth, A. P., & Larson, J. A. (1990). Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys (CSUR)*, *22*(3), 183–236. https://doi.org/10.1145/96602.96604

Silberschatz, A., Sudarshan, S., & Korth, H. F. (2010). *Database System Concepts* (6th Ed.). McGraw-Hill Education.

Stallings, W. (2017). *Cryptography and Network Security, Principles and Practice* (7th Ed.). Pearson.

Tanenbaum, A. S., & Austin, T. (2012). *Structured Computer Organization* (6th Ed.). Pearson.

[i] https://docs.microsoft.com/pt-br/sql/t-sql/data-types/int-bigint-smallint-and-tinyint-transact-sql?view=sql-server-2017