

# Unidade IV

## 7 MELHORES SOLUÇÕES PARA BANCOS DE DADOS

O mercado de sistemas de gerenciamento de bancos de dados operacionais (OPDBMS) é definido por produtos de gerenciamento de banco de dados relacionais e não relacionais, adequados para as transações tradicionais usadas para apoiar processos corporativos. Estes incluem uma ampla variedade de aplicativos de nível empresarial – aplicativos de negócios adquiridos, como ERP e CRM, e sistemas transacionais personalizados. A nossa definição desse mercado também inclui produtos DBMS que apoiam utilizações de processamento de eventos e interações (dados em movimento) para a internet das coisas (*Internet of things* – IoT).

Pode-se definir um DBMS como um sistema de *software* completo usado para definir, criar, atualizar, gerenciar e consultar um banco de dados. O termo "banco de dados" se refere a uma coleção organizada de dados em diversos formatos, que pode ser armazenada em alguma forma de mídia de armazenamento (o que pode incluir unidades de disco rígido, memória *flash*, unidades de estado sólido e DRAM). Um banco de dados precisa incluir mecanismos para isolar e gerenciar exigências de cargas de trabalho, e controlar vários parâmetros do acesso do usuário final nas instâncias gerenciadas dos dados. Além disso, um DBMS deve oferecer interfaces para programas e ferramentas independentes, e permitir e controlar o desempenho de inúmeros tipos de cargas de trabalho simultâneas. Não há pressuposto de que um DBMS deva oferecer suporte ao modelo relacional ou ao conjunto completo de tipos possíveis de dados atualmente em uso. Além disso, não estipulamos que o DBMS deva ser um produto de fonte fechada; produtos de DBMS de código aberto com suporte comercial estão incluídos.

Em algum momento, por alguma particularidade do negócio ou mercado, talvez uma empresa até realize a migração para outro tipo de banco de dados, como o Oracle, SQL Server, PostgreSQL, Firebase, MySQL ou o MongoDB, mas sabemos que isso irá depender de muitas variáveis. Até porque, é impossível migrarmos uma base de dados inteira sem qualquer tipo de planejamento. Seguem alguns SGBDs mais populares:

**Quadro 19 – Os principais SGBDs do mercado mundial**

Rank	DBMS	Database Model
1.	Oracle	Relational, Multi-model
2.	MySQL	Relational, Multi-model
3.	Microsoft SQL Server	Relational, Multi-model
4.	PostgreSQL	Relational, Multi-model
5.	MongoDB	Document, Multi-model

Adaptado de: DB-Engines (s.d.).

### 7.1 Opções de SGBDs proprietários

#### Oracle

Criado por Larry Ellison nos anos 1980, em sua primeira versão ainda bem longe do SGBD que conhecemos hoje, foi inicialmente vendido à base da Força Aérea em Wright-Patterson. Considerada uma das maiores empresas de tecnologia do mundo, a Oracle vem sendo utilizada em grandes empresas e, desde 2009, é também proprietária das linhas de *software* Java.

Por ser dominante no mercado de tecnologia, conhecer a linguagem utilizada no SGBD da Oracle é fundamental para o profissional de tecnologia da informação (TI) que visa ingressar na carreira como um DBA. Como SGBD, possui características de integração no mercado de TI, acompanhando as novidades que vêm surgindo, como a inclusão de multimídia, CAD, gráficos, desenhos, fotografia, textos, som e imagem. Contempla, ainda, desenvolvimentos voltados à programação orientada a objetos e banco de dados semânticos. Implementa a linguagem SQL, inclusive incorporando as modificações realizadas no padrão SQL.



Figura 98 – Logomarca Oracle

#### SQL Server

Lançado em 1988, o poderoso SGBD relacional da Microsoft surge como parte do Windows NT e posteriormente foi comercializado como um produto separado e em constante desenvolvimento. Hoje, é considerado uma plataforma de desenvolvimento que entre suas várias funções permite ao desenvolvedor rapidez e agilidade com flexibilização dos dados, o que lhe dá uma experiência consistente em todas as plataformas e introduz suas inovações no mercado mais rapidamente, já que permite criar aplicativos e implantar em qualquer lugar. Sua inteligência integrada permite o uso de uma plataforma de dados da Microsoft que leva IA para seus dados, possibilitando ganhar conhecimento profundo sobre o seu negócio e clientes.

Seu grande diferencial com relação às outras opções já citadas é a possibilidade de o desenvolvedor utilizar linguagens de programação gerenciadas, como o C# e o Visual Basic .NET, ao invés de usar declarações SQL. Obviamente, também é possível utilizar o SQL, ou melhor, o T-SQL – extensão do SQL (ANSI) para o MS SQL Server. O MS SQL Server também possibilita consultas transparentes.

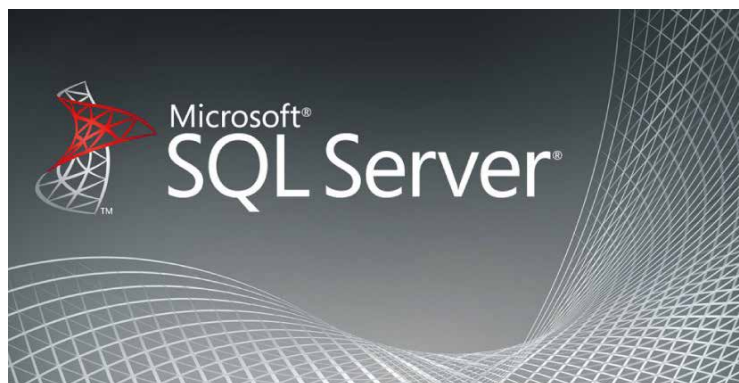


Figura 99 – Logomarca Microsoft SQL Server

### 7.2 Opções de SGBDs gratuitos

#### MySQL

Popularmente conhecido pelos desenvolvedores *open source* (código aberto), é um dos SGBDs mais utilizados e populares, pois sua tecnologia é muito difundida por essa comunidade, o que permite que o desenvolvimento de aplicações seja realizado de acordo com as necessidades de uma organização.

Além disso, sua facilidade de uso e a capacidade de rodar em diferentes sistemas operacionais garante lugar cativo na preferência de grandes empresas por todo o globo. É considerado o SGBD mais popular no mundo, o que pode ser acompanhado pelo *ranking* DB-Engines.



Figura 100 – Logomarca MySQL

#### PostgreSQL

PostgreSQL é outro banco de dados relacional e *open source*, lançado em 1989 e desenvolvido pela PostgreSQL Global Development Group. Com frequência se estabelece entre os cinco SGBDs mais utilizados por organizações do mundo todo.

Assim como MySQL, é muito utilizado para sistemas *web*, permitindo que o desenvolvimento de soluções seja realizado de forma que atenda às necessidades específicas de um negócio, alcançando, dessa forma, um melhor desempenho.



Figura 101 – Logomarca PostgreSQL

### MongoDB

O SGBD MongoDB procura unir o melhor dos sistemas relacionais e as inovações do NoSQL, mantendo muitas características do primeiro, como índices e consultas dinâmicas, mas também com o modelo de dados orientados a documentos.

Dessa forma, os ganhos de agilidade através de esquemas flexíveis e a maior facilidade na escalabilidade horizontal são relevantes quando comparados a outras soluções. O *software* é *open source*, assim como o MySQL e o PostgreSQL, o que permite o desenvolvimento do sistema conforme as necessidades do negócio.

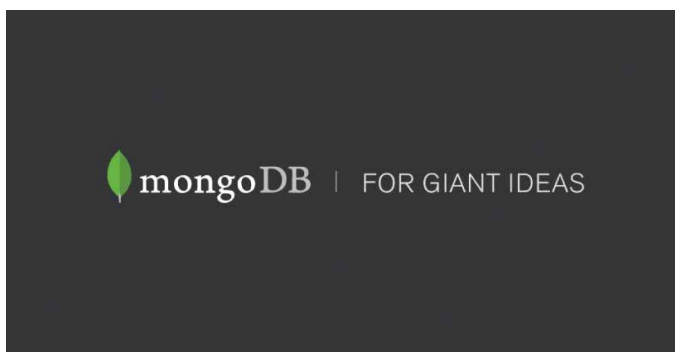


Figura 102 – Logomarca MongoDB

### 7.3 Requisitos de *software* (cliente/servidor)

Antes de iniciarmos a instalação do SQL Server, é necessário efetuar uma verificação do sistema com o SCC (verificador de configuração do sistema), o qual examina o computador em que o SQL Server será instalado. O SCC verifica se existem condições que impedem uma instalação com êxito do SQL Server. Antes que a instalação inicie o assistente de instalação do SQL Server, o SCC recupera o *status* de cada item. Em seguida, compara o resultado com condições necessárias e fornece orientação para a remoção de problemas de bloqueio.



### Lembrete

O verificador da configuração do sistema gera um relatório que contém uma breve descrição de cada regra executada, bem como o *status* de execução. É possível acessá-lo nos Documentos do SQL, no *site* da Microsoft.

O verificador da configuração do sistema gera um relatório que contém uma breve descrição de cada regra executada, bem como o *status* de execução. Segundo o *site* da Microsoft, estas são as configurações de *hardware*:

**Quadro 20 – Requisitos de *hardware***

Componente	Requisito
Disco rígido	O SQL Server requer no mínimo 6 GB de espaço disponível no disco rígido. Os requisitos de espaço em disco variam de acordo com os componentes do SQL Server instalados.
Monitoramento	O SQL Server requer um monitor com resolução Super-VGA (800 x 600) ou superior.
Internet	A funcionalidade de internet requer acesso à internet (a cobrança de taxas poderá ser aplicável).
Memória	Mínimo: Edições Express: 512 MB Todas as outras edições: 1 GB Recomendado: Edições Express: 1 GB Todas as outras edições: pelo menos 4 GB e deve ser aumentado à medida que o tamanho do banco de dados aumenta, para garantir um ótimo desempenho.
Velocidade do processador	Mínimo: processador x64: 1,4 GHz Recomendado: 2,0 GHz ou mais rápido
Tipo de processador	Processador x64: AMD Opteron, AMD Athlon 64, Intel Xeon com suporte Intel EM64T, Intel Pentium IV com suporte EM64T

Adaptado de: Microsoft (2020).

Os requisitos de memória e processador a seguir aplicam-se a todas as edições do SQL Server.

### Requisitos de *software*

Os requisitos de instalação variam de acordo com as necessidades do seu aplicativo. As diferentes edições do SQL Server acomodam os requisitos exclusivos de desempenho, tempo de execução, preço de organizações e indivíduos. Os componentes do SQL Server que serão instalados também dependem de seus requisitos específicos.

Os seguintes requisitos de *software* se aplicam a todas as instalações:

**Quadro 21 – Requisitos de *software***

Componente	Requisito
Sistema operacional	Windows 10 TH1 1507 ou superior Windows Server 2016 ou superior
.NET Framework	Os sistemas operacionais mínimos incluem o .NET Framework.
Software de rede	Os sistemas operacionais com suporte para SQL Server têm <i>software</i> de rede interno. As instâncias nomeadas e padrão de uma instalação autônoma são compatíveis com os seguintes protocolos de rede: memória compartilhada, pipes nomeados e TCP/IP.

Adaptado de: Microsoft (2020).

### 7.4 Tipos de licença (GNU GPL, BSD)

Licenças de *software* fornecem (e limitam) o direito de uso de uma aplicação ao usuário final, que pode ser uma pessoa, uma empresa ou uma entidade governamental. Ela define quais serão os protocolos de uso e as regras sobre cópias e alterações do código-fonte, bem como punições aplicáveis no caso de descumprimento. Os licenciamentos também delimitam o suporte ao usuário, as políticas de atualização e a garantia de serviços. Existem dois modelos para licenciar programas de *software*: o modelo proprietário e o modelo *open source* – também conhecido como de código aberto. Tais modelos se distinguem em relação aos direitos de posse, à modificação e à replicação.

#### GNU *General Public License* (GNU GPL)

Conhecida em português como GNU licença de uso geral, a GNU GPL (ou só GPL) é um dos principais modelos de licenciamento de *software* de código aberto do mercado, garantindo ao usuário final os direitos de:

- Executar o sistema.
- Estudar seu código-fonte.
- Modificar o código-fonte.
- Compartilhar livremente o programa com a comunidade.

Atualmente mantida pela Free Software Foundation, a GNU GPL está em sua terceira versão – também conhecida como GNU GPLv3. Essa foi uma das primeiras licenças disponibilizadas por meio do modelo de livre direito de cópia (*copyleft*). Isso quer dizer que, com esse tipo de licença, todo o trabalho desenvolvido a partir de um código originalmente distribuído em GNU GPL pode ser redistribuído.

As licenças GPL possuem uma cláusula de uso opcional que garante que o sistema terá seu contrato de uso automaticamente atualizado para a última versão do GNU GPL. Dessa forma, o usuário pode optar por manter o modelo de licenciamento original ou migrar para a nova versão, se julgar necessário.

## Berkeley Software Distribution (BSD)

A licença BSD faz parte de um grupo de licenças voltadas para programas de código aberto, diferentes da GNU GPL. Conhecidas como licenças permissivas, essa categoria dá mais liberdade em relação à forma com que o desenvolvedor trabalha o código de terceiros. Enquanto na licença GPL há regras para que o novo *software* herde o modelo de licenciamento do código original, isso não é obrigatório na licença BSD.

Por mais que os direitos sejam semelhantes aos dados pelas licenças GPL, nesse caso, o desenvolvedor pode modificar o tipo de modelo de distribuição do código-fonte original ou mesmo não o distribuir. Isso permite que sistemas proprietários incorporem linhas de códigos de origem *open source*, como acontece no Windows e macOS. Para liberar o uso, basta que a fonte original seja citada.

## 7.5 Análise de custo/benefício

A escolha da edição do SQL Server 2019 para oferecer suporte ao aplicativo da camada de dados pode parecer uma tarefa simples, mas, de fato, é preciso dedicar algum tempo para pensar nessa decisão e consultar as partes interessadas do negócio e outros departamentos de TI para trazer suas opiniões sobre essa escolha. A primeira coisa a considerar é que existem cinco edições do SQL Server. Essas edições não apenas têm níveis diferentes de funcionalidade, mas também possuem considerações de licença diferentes. Além disso, da perspectiva de um suporte operacional, podemos achar que o TCO (custo total de propriedade) da propriedade aumenta se você permitir que aplicativos da camada de dados sejam hospedados em versões do SQL Server que não sejam implantadas estrategicamente em sua propriedade.

Uma discussão completa sobre recursos e considerações de licenciamento está além do escopo deste livro-texto; no entanto, os quadros a seguir detalham os modelos de licenciamento disponíveis para cada edição do SQL Server e o objetivo principal de cada edição.

**Quadro 22 – Modelos de licenciamento de edições SQL Server**

Edição	Modelo de licença	Comentários
Enterprise	Por núcleo	–
Standard	Por núcleo Server + CAL	–
Web	Somente hospedagem de terceiros	–
Developer	Free (uso não comercial)	Uso não comercial
Express	Edição gratuita do SQL Server	Funcionalidade limitada e pequenos limites de capacidade, como um tamanho de banco de dados de 10 GB, um limite de 1 GB na RAM e um limite de CPU de um soquete ou quatro núcleos

Adaptado de: Microsoft (2020).

Uma CAL é uma licença de acesso de cliente, na qual um cliente pode se referir a um usuário ou dispositivo. É possível optar por comprar licenças de usuário ou dispositivo com base nas quais serão mais baratas para o seu ambiente. Por exemplo, se sua organização tiver um SQL Server que ofereça suporte a um *call center* que tinha 100 computadores e funcionava 24 horas por dia, 7 dias por semana, com três turnos de 8 horas, então você teria 100 dispositivos e 300 usuários, portanto as CALs de dispositivo seriam a opção mais sensata para escolher.

Para resumir, se você tiver mais usuários que dispositivos, deverá escolher o dispositivo CALs. Se você possui mais dispositivos que usuários, por outro lado, deve escolher usuário CALs. A Microsoft também fornece uma ferramenta chamada Microsoft Assessment and Planning (MAP). Um *kit* de ferramentas para SQL Server, que o ajudará a planejar seus requisitos de licenciamento.

**Quadro 23 – Visão geral das edições SQL Server**

Edição	Visão geral da edição
Enterprise	Edição completa dos sistemas SQL Server for Enterprise e aplicativos críticos.
Standard	Banco de dados principal e funcionalidade de BI, voltada para sistemas de nível departamental e aplicativos não críticos.
Web	Está disponível apenas para provedores de serviços que hospedam <i>sites</i> públicos que usam o SQL Server.
Developer	Uma edição completa, no nível da edição Enterprise, mas destinada ao desenvolvimento e não permitido para uso em sistemas de produção.
Express	Uma versão gratuita e básica do SQL Server voltada para pequenos aplicativos com requisitos de dados locais.

Adaptado de: Microsoft (2020).

As versões do SQL Server escolhidas irão oferecer suporte em sua empresa, os aplicativos irão variar, dependendo dos requisitos do projeto, da organização, requisitos e infraestrutura subjacente. Por exemplo, se a sua organização hospeda toda a propriedade do SQL Server em uma nuvem privada, é provável que tenha suporte apenas o Enterprise Edition, pois você licenciará a infraestrutura subjacente.

Como alternativa, se sua organização utiliza predominantemente caixas físicas, será necessário, provavelmente, oferecer suporte a uma mistura de versões do SQL Server, como Enterprise e Standard Edition. Isso dará aos projetos a flexibilidade de reduzir seus custos, se eles apenas requerem um subconjunto de recursos e não esperam cargas de trabalho de alto volume e, portanto, podem conviver com os limites que a Standard Edition impõe à RAM e à CPU.

A próxima coisa que deve ser considerada, antes de escolher qual edição será usada, é se será utilizada ou não uma instalação do Windows Server Core do SQL Server. As instalações no Server Core podem ajudar a melhorar a segurança, reduzindo a superfície de ataque de seu servidor. O Server



Core é uma instalação mínima, portanto há menos superfície para atacar e menos vulnerabilidades de segurança. Também pode melhorar o desempenho, porque não possui sobrecarga da GUI e porque muitos aplicativos de uso intensivo de recursos não podem ser instalados. Se a decisão for por usar o Server Core, também será importante entender os impactos de fazê-lo.

Da perspectiva do SQL Server, os seguintes recursos não podem ser usados:

- Reporting Services.
- SQL Server Data Tools (SSDT).
- Client Tools Backward Compatibility.
- Client Tools SDKI.
- SQL Server Books Online.
- Distributed Replay Controller.
- Master Data Services (MDS).
- Data Quality Services (DQS).

Os seguintes recursos podem ser usados, mas apenas de um servidor remoto:

- Management Tools.
- Distributed Replay Client.

Para uma perspectiva mais ampla do suporte operacional, é preciso garantir que todas as suas equipes operacionais (DBAs, operações do Windows etc.) estejam em condições de oferecer suporte ao Server Core. Por exemplo, se sua equipe de DBA depende muito de terceiros, ferramenta gráfica para interrogar os planos de execução, isso precisa ser instalado localmente no servidor? Existe uma ferramenta alternativa que atenda às suas necessidades? A partir da perspectiva de operações do sistema operacional Windows, a equipe possui as ferramentas para monitorar remotamente e gerenciar o servidor? Existem ferramentas de terceiros com as quais eles trabalham e que precisem ser substituídas?

Você também deve considerar se a sua equipe de operações possui o conjunto de habilidades para gerenciar sistemas usando processos de linha de comando. Caso contrário, considere que um treinamento ou qualificação superior podem ser necessários.



### Observação

Para determinar a versão do SQL Server em uso no seu dispositivo, você pode usar qualquer um dos métodos a seguir.

**Método 1:** conecte-se ao servidor usando o Explorador de Objetos no SQL Server Management Studio. Depois que o Explorador de Objetos estiver conectado, ele mostrará as informações de versão entre parênteses, juntamente com o nome de usuário usado para se conectar à instância específica do SQL Server.

**Método 2:** veja as primeiras linhas do arquivo Errorlog para essa instância. Por padrão, o *log* de erros está localizado em Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\LOG\ERRORLOG e arquivos *ERRORLOG.n*.

## 8 DIMENSIONAMENTO DE HARDWARE

Ao planejar os requisitos de *hardware* para o servidor, o ideal é que se faça o exercício de planejamento de capacidade total para, assim, estimar os requisitos de *hardware* e *software* que serão suportados pelo servidor. Ao realizar esse exercício, certifique-se de levar em consideração o ciclo de vida do *hardware* padrão da sua empresa, em vez de planejar apenas por hoje. Dependendo da sua organização, isso pode ser entre 1 e 5 anos, mas geralmente será de 3 anos.

Isso é importante para evitar subdimensionar ou superdimensionar o servidor. Equipes de projetos, geralmente, precisarão sobrecarregar os servidores para garantir o desempenho. Essa abordagem não só é custosa à empresa, mas, em alguns ambientes, ela pode realmente ter um efeito prejudicial no desempenho. Um exemplo disso seria uma infraestrutura de nuvem privada com recursos compartilhados. Nesse cenário, servidores de tamanho grande podem ter um impacto negativo em todo o ambiente, incluindo o servidor de grandes dimensões.

### 8.1 Especificando requisitos mínimos estratégicos

Ao especificar os requisitos mínimos de *hardware* para o SQL Server dentro do seu ambiente, pode-se optar por especificar os requisitos mínimos para instalar o SQL Servidor - 4 GB de RAM e uma única CPU de 2 GHz (com base na edição Enterprise). No entanto, pode ser mais útil pensar em suporte operacional dentro de sua empresa.

Por exemplo, se o ambiente consistir predominantemente em uma infraestrutura de nuvem privada, convém especificar no mínimo 2 Cores e 4 GB de RAM + (número de núcleos \* 1 GB), pois isso pode estar de acordo com os padrões da sua empresa.

Por outro lado, se você tem uma empresa com uma diversidade de localização em que seu parque de máquina está distribuído de forma heterogênea e que está crescendo ultimamente, é possível planejar como os projetos usados em seu parque de *hardware* utilizarão o compartilhado do SQL Server, optando por impor especificações mínimas muito mais altas, como 32 GB de RAM e 2 soquetes/4 núcleos. O raciocínio aqui é que em qualquer projeto sem grande rendimento os requisitos seriam “forçados” a usar seu parque compartilhado para evitar os custos pesados associados a um sistema desnecessariamente grande.



### Lembrete

*Storage Area Network* (rede dedicada para armazenamento), ou apenas SAN, são três palavras que podem causar receio a um administrador de banco de dados. O DBA deve compreender os conceitos dos serviços SAN e virtualização. Além de representar uma mudança fundamental, eles também facilitam o gerenciamento geral e reduzem o custo total de propriedade (TCO).

### Dimensionamento dos *links* de comunicação

O armazenamento é uma consideração muito importante para qualquer instalação do SQL Server. As próximas seções discutirão armazenamento conectado localmente e armazenamento SAN (rede de área de armazenamento), bem como considerações para posicionamento de arquivo.

## 8.2 Armazenamento local

Se a localização de seus serviços for do tipo armazenamento local, considere cuidadosamente o arquivo *layout*. Por sua natureza, o SQL Server, geralmente, é vinculado de entrada e saída (IO), portanto, configurar o subsistema de E/S é um dos aspectos críticos para o desempenho.

Primeiro, é preciso separar os arquivos de dados e os arquivos de *log* dos bancos de dados do usuário em discos ou matrizes separados, e também separar o TempDB, que é o banco de dados do sistema mais usado. Se todos esses arquivos residem em um único volume, é provável que ocorra uma contenção de disco enquanto o SQL Server tenta gravar em todos eles ao mesmo tempo.

Normalmente, o armazenamento local será apresentado ao seu servidor como RAID (matriz redundante de discos baratos) e vários níveis de RAID estão disponíveis. Existem muitos níveis de RAID disponíveis, mas os mais comuns estão descritos nas seções a seguir, juntamente com suas vantagens e desvantagens. Isso ajudará a selecionar o nível RAID mais adequado, com o equilíbrio mais adequado entre desempenho e tolerância a falhas.

### RAID 0

Um volume RAID 0 consiste em entre dois ou n eixos, em que os *bits* de dados são distribuídos em todos os discos da matriz. Isso fornece excelente desempenho, no entanto, não fornece tolerância a falhas. A perda de qualquer disco dentro da matriz significa que toda a matriz falhará. Isso é ilustrado na figura a seguir.

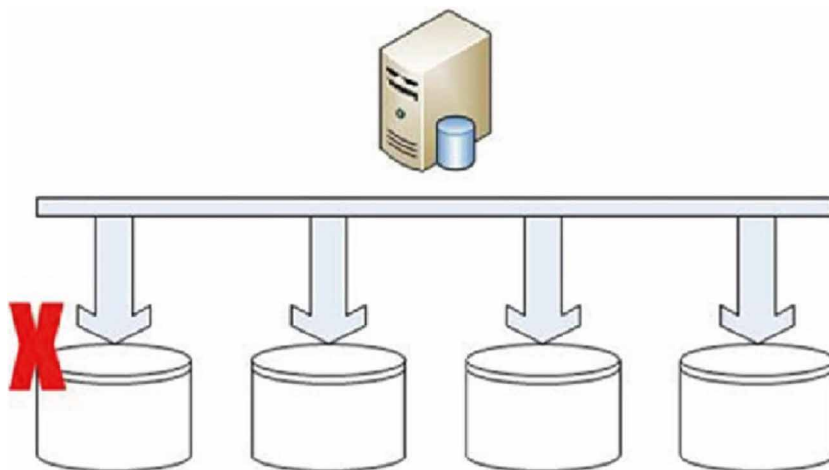


Figura 103 – A matriz RAID 0 não fornece redundância

### RAID 1

Um volume RAID 1 consistirá em dois eixos, trabalhando juntos como um par espelhado. Este fornece redundância em caso de falha de um dos eixos-árvore, mas ocorre no custo do desempenho de gravação, porque todas as gravações no volume precisam ser feitas duas vezes. Este método de redundância é ilustrado na próxima figura.

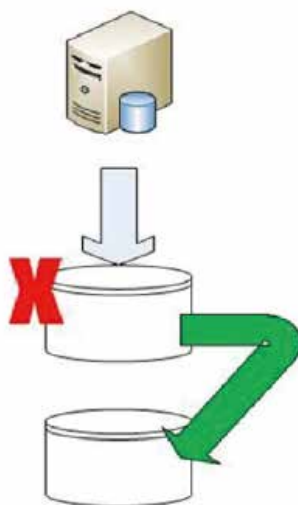


Figura 104 – A matriz RAID 1 fornece redundância de espelhamento

## RAID 5

Um volume RAID 5 consistirá em três ou  $n$  eixos e fornece redundância de exatamente um disco dentro da matriz. Como os blocos de dados são distribuídos por vários eixos, o desempenho de leitura do volume será muito bom, mas, novamente, neste tipo de serviço é dispensada uma alta disponibilidade de gravação. O desempenho da gravação é prejudicado, pois a redundância necessita de distribuição de *bits* de paridade em todos os eixos na matriz. Isso significa que existe uma penalidade de desempenho de quatro gravações para cada gravação no volume. Isso é independente do número de discos na matriz. A razão para esta penalidade é arbitrária, pois a paridade dos *bits* é distribuída da mesma maneira que os dados. O controlador lerá os dados originais e a paridade original e, em seguida, escreverá os novos dados e a nova paridade, sem precisar ler todos os outros discos na matriz. Este método de redundância é ilustrado na figura seguinte. Vale ressaltar, no entanto, que, se um eixo-árvore dentro da matriz falhar, o desempenho será visivelmente prejudicado. A reconstrução em disco a partir da paridade de *bits* contidos em seus pares pode levar um longo período de tempo, especialmente para um disco com uma grande capacidade.

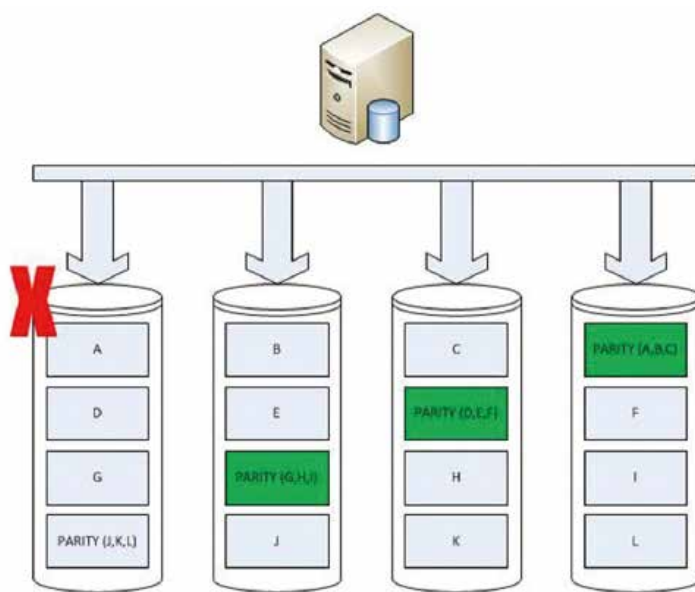


Figura 105 – A matriz RAID 5 fornece redundância através de *bits* de paridade

## RAID 10

Um volume RAID 10 consistirá em quatro a  $n$  discos, mas sempre será um número par. Isto fornece a melhor combinação de redundância e desempenho. Funciona criando um faixa de espelhos. Os *bits* são distribuídos, sem paridade, em metade dos discos como são para o RAID 0, mas são espelhados para a outra metade dos discos na matriz.

Isso é conhecido como nível RAID aninhado ou híbrido e significa que metade dos discos dentro da matriz podem ser perdidos, desde que nenhum dos discos com falha esteja dentro do mesmo par espelhado. Isso é ilustrado na figura a seguir.

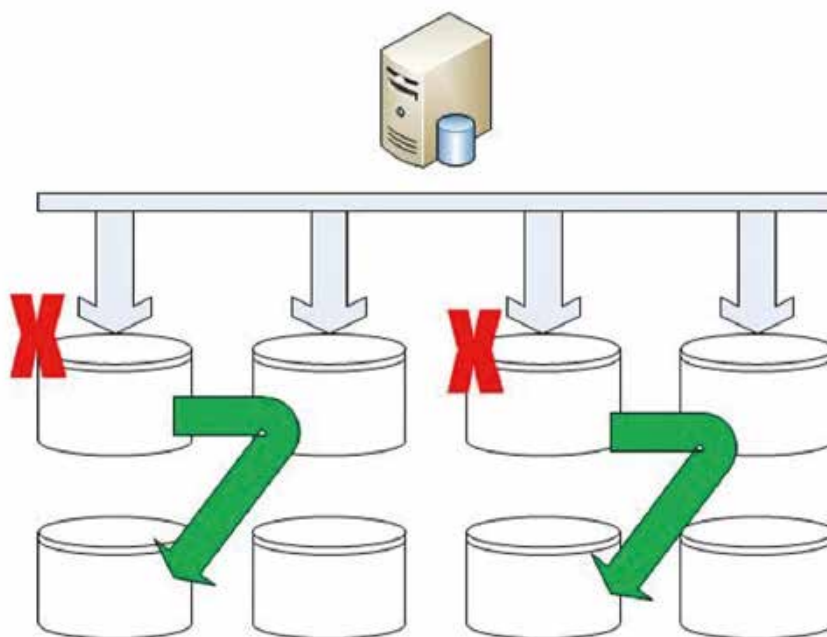


Figura 106 – A matriz RAID 10 fornece redundância de espelhamento de cada disco na faixa

### 8.3 Posicionamento do arquivo

É geralmente aceito que o RAID 0 não deve ser usado para nenhum arquivo do SQL Server, há profissionais que sugerem que o RAID 0 pode ser aceitável para arquivos TempDB. A justificativa aqui é que um TempDB muito usado requer desempenho muito rápido, sendo este recriado toda vez que a instância é reiniciada, não requer redundância.

Sua instância do SQL Server precisa do TempDB para funcionar. Se o TempDB for perdido, sua instância será desativada e, se o TempDB não puder ser recriado, não será possível trazer sua instância de volta. Portanto, se o TempDB for hospedado em uma matriz RAID 0 e um dos discos nessa matriz falhar, não será possível trazer a instância de volta até que se execute uma das seguintes ações:

- 1) Aguarde a equipe de armazenamento colocar o *array* RAID 0 novamente *on-line*.
- 2) Inicie a instância no "modo de configuração mínima" e use SQLCMD para alterar a localização do TempDB.

Quando uma dessas etapas estiver concluída, pode-se descobrir que as partes interessadas estão pulando para cima e para baixo, portanto, é melhor evitar essa opção. Por esta razão, o TempDB, geralmente, é melhor colocado em uma matriz RAID 10, sempre que possível. Esta fornecerá o melhor nível de desempenho para o banco de dados, porque seu tamanho é significativamente menor que os arquivos do banco de dados do usuário e não terá o mesmo nível de custo-implicação.

Se seus bancos de dados estiverem usando apenas os recursos básicos do SQL Server, provavelmente será encontrado esse RAID 1, que é uma boa opção para seus arquivos de *log*. O RAID 5 geralmente não é adequado, devido à natureza intensiva de gravação do *log* de transações. Em alguns casos, recomenda-se até começar com o RAID 1 para executar melhor que o RAID 10 para o *log* de transações. Isto é devido a natureza sequencial da atividade de gravação.



### Resumo

Nesta unidade, foram apresentadas as melhores soluções para sistemas gerenciadores de banco de dados (SGBDs), como o Oracle, PostgreSQL, MySQL, Microsoft SQL Server e MongoDB, bem como suas principais características, opções de SGBDs proprietários e gratuitos, além dos requisitos de *software* (cliente/servidor).

Apresentamos, ainda, os tipos de licenças comuns: GNU General Public License (GNU GPL), conhecida por ser um dos principais modelos de licenciamento de *software* de código aberto do mercado; e *Berkeley Software Distribution* (BSD), que faz parte de um grupo de licenças voltadas para programas de código aberto.

Também apresentamos e avaliamos qual o tipo de planejamento de requisitos de *hardware* para o servidor é o ideal para cada organização, levando em consideração o ciclo de vida do *hardware* padrão da empresa e a necessidade da organização.



### Exercícios

**Questão 1.** Com relação aos procedimentos de *backup* de um banco de dados, considere as afirmativas a seguir:

I – Um dos principais objetivos da execução de *backups* em um banco de dados é evitar que dados sejam permanentemente perdidos, seja devido a falhas de *hardware*, seja devido a falhas de *software* ou mesmo falhas de segurança.

II – Podemos utilizar *backups* para a replicação dos ambientes de produção, o que é feito, por exemplo, para a elaboração de testes. A existência de um ambiente de teste igual ao ambiente de produção permite que os testes feitos em um *software* sejam mais realistas e abrangentes, sem arriscar danificar dados ou interromper a rotina de uma empresa.

III – É importante que sejam feitos *backups* completos de um banco de dados, e em muitos setores nos quais a segurança desses dados é absolutamente fundamental, cópias são mantidas em locais

separados e distantes dos servidores que o originaram, para garantir a possibilidade de recuperação mesmo no caso de uma catástrofe.

É correto o que se afirma em:

A) I, apenas.

B) II, apenas.

C) III, apenas.

D) I e II, apenas.

E) I, II e III.

Resposta correta: alternativa E.

### Análise das afirmativas

I – Afirmativa correta.

Justificativa: diversos tipos de falhas podem levar à perda de dados em um servidor. Falhas de *hardware* são causas muito comuns para a perda de dados. Contudo, falhas no *software* também podem ocasionar perda de dados, por exemplo devido a erros (comumente chamados de *bugs*) em uma nova versão de uma aplicação. Além disso, um grande problema na atualidade refere-se às falhas de segurança, que podem permitir que invasores alterem ou destruam dados importantes. Esses são apenas alguns dos motivos que indicam a importância de uma estratégia de *backup* eficiente.

II – Afirmativa correta.

Justificativa: os testes de uma aplicação são fundamentais para garantir um produto de qualidade para o usuário final. É importante que exista um ambiente de testes que seja o mais fiel possível ao ambiente de produção, permitindo uma ampla gama de testes em situações realistas. *Backups* podem ser utilizados para criar um espelho do ambiente de produção, aumentando a confiabilidade dos testes.

III – Afirmativa correta.

Justificativa: *backups* completos costumam levar mais tempo para serem feitos e consomem mais recursos, especialmente espaço. Contudo, eles são fundamentais para se reestabelecer um sistema no caso de uma falha catastrófica.



**Questão 2.** Considere as afirmativas a seguir sobre as formas de acesso a um banco de dados:

I – A utilização da API ODBC permite que uma aplicação acesse diferentes bancos de dados de forma uniforme e independente dos detalhes de um SGBD específico.

II – A linguagem Java apresenta uma API de acesso a banco de dados chamada de JDBC. A utilização das chamadas pontes JDBC-para-ODBC permite que uma aplicação escrita na linguagem Java acesse diferentes tipos de banco de dados sem a necessidade da utilização de uma API específica.

III – Na linguagem Java, é possível utilizar JDBC para executar o comando SELECT da linguagem SQL em um banco de dados, mas não é possível utilizar o comando INSERT.

É correto o que se afirma em:

A) I, apenas.

B) II, apenas.

C) III, apenas.

D) I e II, apenas.

E) I, II e III.

Resposta correta: alternativa D.

### Análise das afirmativas

I – Afirmativa correta.

Justificativa: um dos objetivos da criação da API ODBC foi justamente garantir uma independência entre a aplicação e o SGBD, criando uma interface de acesso uniforme.

II – Afirmativa correta.

Justificativa: a linguagem Java também apresenta uma API de acesso de banco de dados similar a ODBC, chamada de JDBC. Existem diversas formas de uso, uma delas chamada de ponte JDBC-para-ODBC.

III – Afirmativa incorreta.

Justificativa: é possível executar ambos os comandos, tanto SELECT quanto INSERT, através da API JDBC.

## FIGURAS E ILUSTRAÇÕES

### Figura 1

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 205.

### Figura 6

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e administração*. 8. ed. São Paulo: Cengage, 2011. p. 116.

### Figura 7

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 25.

### Figura 8

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 26.

### Figura 9

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 26.

### Figura 10

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 27.

### Figura 11

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 27.

### Figura 12

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 28.

### **Figura 13**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 29.

### **Figura 14**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 30.

### **Figura 15**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 31.

### **Figura 16**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 67.

### **Figura 17**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 69.

### **Figura 18**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 69.

### **Figura 19**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 71.

### **Figura 20**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 147.

### **Figura 22**

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011. p. 154.

### **Figura 23**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 158.

### **Figura 24**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 158.

### **Figura 25**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 136.

### **Figura 26**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 158.

### **Figura 29**

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e administração*. 8. ed. São Paulo: Cengage, 2011. p. 140.

### **Figura 30**

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e administração*. 8. ed. São Paulo: Cengage, 2011. p. 120.

### **Figura 32**

SANCHES, A. R. *Modelos entidade relacionamento*. 2005. Disponível em: <https://bit.ly/2Pp4Ydx>. Acesso em: 26 abr. 2021.

### **Figura 33**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 340.

### **Figura 34**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 341.

### **Figura 35**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 49.

### **Figura 36**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 49.

### **Figura 37**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 353.

### **Figura 38**

TEOREY, T. *et al. Projeto e modelagem de banco de dados*. Rio de Janeiro: Elsevier, 2014. p. 234.

### **Figura 39**

TEOREY, T. *et al. Projeto e modelagem de banco de dados*. Rio de Janeiro: Elsevier, 2014. p. 235.

### **Figura 59**

IMAGE002.JPG. Disponível em: <https://bit.ly/3gEAorq>. Acesso em: 26 abr. 2021.

### **Figura 62**

IMAGE004.PNG. Disponível em: <https://bit.ly/3ewz7jF>. Acesso em: 26 abr. 2021.

### **Figura 64**

IMAGE006.JPG. Disponível em: <https://bit.ly/2QUZsQa>. Acesso em: 26 abr. 2021.

### **Figura 66**

IMAGE008.JPG. Disponível em: <https://bit.ly/3nIO4ZI>. Acesso em: 26 abr. 2021.

### **Figura 67**

SALLAI, F. *SQL JOIN: entenda como funciona o retorno dos dados*. 2014. Disponível em: <https://bit.ly/3viXBn8>. Acesso em: 26 abr. 2021.

### **Figura 68**

IMAGE014.JPG. Disponível em: <https://bit.ly/3xqAvwN>. Acesso em: 26 abr. 2021.

### **Figura 98**

ORACLE\_REDLOGO.JPG. Disponível em: <https://bit.ly/33WUaXI>. Acesso em: 20 maio 2021.

### **Figura 99**

MICROSOFT-SQL-SERVER-LOGO.Q3XD. Disponível em: <https://bit.ly/2SeAoUI>. Acesso em: 20 maio 2021.

### **Figura 100**

LOGO-MYSQL-170X115.PNG. Disponível em: <https://bit.ly/3xqb8Lw>. Acesso em: 27 abr. 2021.

### **Figura 101**

POSTGRESQL\_ELEPHANT.SVG. Disponível em: <https://bit.ly/3uaeG2F>. Acesso em: 27 abr. 2021.

### **Figura 102**

MONGODB-FOR-GIANT-IDEAS-BBAB5C3CF8.PNG. Disponível em: <https://bit.ly/3eyuuFL>. Acesso em: 27 abr. 2021.

### **Figura 103**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 417.

### **Figura 104**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 417.

### **Figura 105**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 417.

### **Figura 106**

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011. p. 417.

## REFERÊNCIAS

### Textuais

- AMARAL, F. *Introdução à ciência de dados*. Rio de Janeiro: Alta Books, 2014.
- BOGGIANO, T.; FRITCHEY, G. *Query Store for SQL Server 2019: identify and fix poorly performing queries*. [s.l.]: Apress, 2019.
- CARTER, P. A. *Pro SQL Server 2019 Administration: a guide for the modern DBA*. [s.l.]: Apress, 2019.
- DATE, C. J. *Introdução a sistemas de bancos de dados*. Rio de Janeiro: Campus, 1991.
- DB-ENGINES. *Knowledge Base of Relational and NoSQL Database Management Systems*. [s.d.]. Disponível em: <https://bit.ly/3f2gUM4>. Acesso em: jun. 2020.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados: fundamentos e aplicações*. 4. ed. São Paulo: Pearson, 2011.
- FERNANDES, L. *Oracle 9i para desenvolvedores: curso completo*. Rio de Janeiro: Axcel Books, 2002.
- FERREIRA, L. *Linguagem SQL: guia prático de aprendizagem*. São Paulo: Érica, 2011.
- GENNICK, J.; LUERS, T. *Aprenda em 21 dias PL/SQL*. 2. ed. Rio de Janeiro: Campus, 2000.
- GONÇALVES, E. *PL-SQL: domine a linguagem do banco de dados Oracle*. São Paulo: Casa do Código, 2015.
- HEUSER, C. A. *Projeto de banco de dados*. 4. ed. Porto Alegre: Sagra Luzzatto, 2001.
- LUCIDCHART. *O que é um diagrama entidade relacionamento?* [s.d.]. Disponível em: <https://bit.ly/3uzFfxL>. Acesso em: 7 abr. 2021.
- MACHADO F.; ABREU, M. *Projeto de banco de dados: uma visão prática*. 17. ed. São Paulo: Érica, 2012.
- MANZANO, J. N. G. *SQL Server 2008-Express: guia prático*. São Paulo: Érica, 2009.
- MICROSOFT. *Instalar o MAP Toolkit*. [s.d.]. Disponível em: <https://bit.ly/3gegseP>. Acesso em: 14 abr. 2021.
- MICROSOFT. *SQL Server 2016 e 2017: requisitos de software e de hardware*. 2020. Disponível em: <https://bit.ly/2REk3IU>. Acesso em: 14 abr. 2021.
- MICROSOFT. *Tipos de dados (Transact-SQL)*. 2017. Disponível em: <https://bit.ly/3vfl1cU>. Acesso em: 26 abr. 2021.

RAMAKRISHNAN, R.; GEHRKE, E. J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2011.

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e administração*. 8. ed. São Paulo: Cengage, 2011.

TEOREY, T. *et al. Projeto e modelagem de banco de dados*. Rio de Janeiro: Elsevier, 2014.

THORWALD, G. A. G. *UML2: uma abordagem prática*. 2. ed. São Paulo: Novatec, 2011.





Handwriting practice lines consisting of 30 horizontal blue lines. Each line is preceded by a small blue dot on the left margin, serving as a guide for letter height and placement.



A series of horizontal lines for writing, consisting of 30 evenly spaced lines across the page.





# Interativa

Informações:  
[www.sepi.unip.br](http://www.sepi.unip.br) ou 0800 010 9000