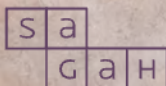


# ADMINISTRAÇÃO DE BANCO DE DADOS

Claudia Abreu Paes



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS



# Inicialização e desativação de um banco de dados

## Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Determinar as características de inicialização e desativação de um SGDB.
- Identificar erros na inicialização e/ou desativação.
- Resolver os erros na inicialização e/ou desativação.

## Introdução

A administração de um banco de dados é responsável por desenvolver e administrar, de modo centralizado, as estratégias, os procedimentos e as práticas para a gerência dos recursos de dados e aplicativos. A gerência de recursos é desempenhada pelo administrador de banco de dados (*DataBase Administrator* [DBA]). Ela implica o acompanhamento da execução, buscando sempre o bom desempenho, a segurança e a integridade das informações.

O acompanhamento pode ser realizado de várias formas, por meio de comandos *on-line* de análise, verificação de *logs* (registros de execução) ou mesmo observação de resultados dos processamentos de uso do banco de dados. Além disso, vários são os recursos geridos pelo DBA, como inicialização e desativação de banco de dados, gerenciamento de espaço utilizado e livre, gerenciamento de usuários, gerenciamento de privilégios e desempenho.

Todo banco de dados deve estar disponível para uso. Na verificação de qualquer anomalia ou necessidade de reparos (sejam preventivos ou corretivos), o administrador deve desativar os recursos do banco para garantir a segurança e a integridade dos dados armazenados.

Neste capítulo, você vai estudar os procedimentos de inicialização e desativação dos bancos de dados, suas particularidades, possíveis erros e soluções.

## Inicialização e desativação de um SGBD

Existem muitos sistemas gerenciadores de banco de dados (SGBDs). Cada um deles possui determinada estrutura e disponibiliza a instrução para a inicialização e a desativação de um banco de dados. Aqui, para exemplificar os processos, é utilizado o SGBD MySQL, fornecido pela empresa Oracle.

Todo SGBD possui seus mecanismos de inicialização e desativação, o que disponibiliza ou indisponibiliza recursos para o uso. Quando o servidor que hospeda o banco de dados é iniciado, normalmente o procedimento que torna o banco de dados disponível para a conexão dos usuários já está incluído em sua rotina de inicialização. Contudo, o DBA pode, via console, realizar as atividades de ativação e desativação de um banco de dados, como iniciar, interromper, pausar e reiniciar.



### Link

O SGBD MySQL é um sistema de gerenciamento de banco de dados de código aberto, fornecido sob a licença GPL, que dá direito a compartilhar e alterar o *software* livre, garantindo a gratuidade para todos os usuários. Para instalar o MySQL, acesse o *link* a seguir.

<https://qr.go.page.link/hdTS>

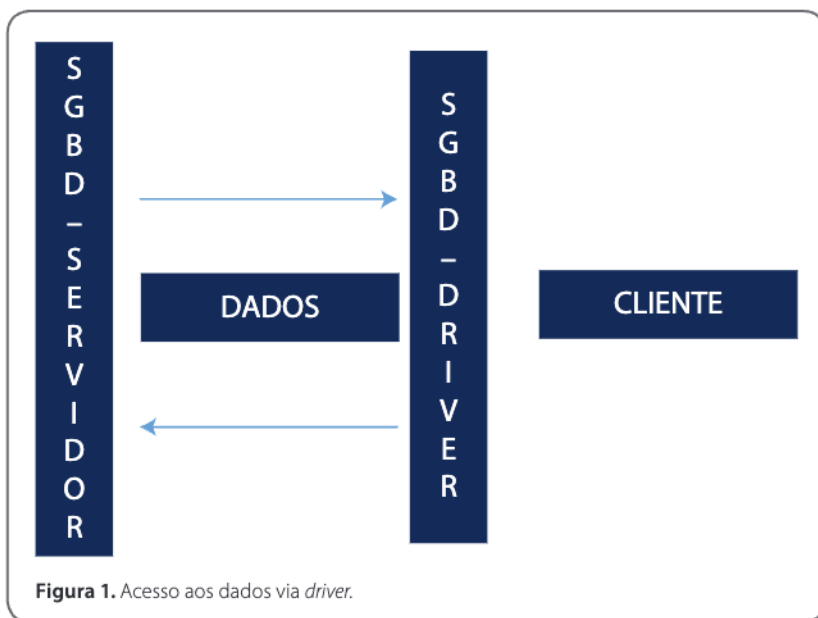
Um banco de dados possui estados que representam a situação para a sua utilização em determinado momento. A seguir, veja quais são os estados possíveis.

- *On-line*: o banco de dados está disponível para utilização pelos usuários.
- *Off-line*: o banco de dados está indisponível. Esse estado é normalmente utilizado temporariamente pelo administrador do banco de dados para realizar uma manutenção sem a interferência de ações dos usuários.

- *Restoring*: o banco de dados está indisponível porque um ou mais arquivos da sua estrutura estão sendo restaurados.
- *Recovering*: o banco de dados está indisponível por estar sendo recuperado. O processo de recuperação é temporário e, ao ser finalizado com sucesso, retorna o banco de dados para o estado *on-line*. Caso ocorra algum problema no processo de recuperação, o banco de dados fica no estado *suspect*.
- *Recovering\_pending*: o processo de recuperação não pode ser concluído por uma questão de recursos do ambiente. É necessária a interferência do administrador do banco para resolver a questão e dar continuidade ao processo de recuperação.
- *Suspect*: o banco de dados está indisponível porque há dúvidas quanto à integridade de um ou mais de seus arquivos. Novamente, é necessária uma intervenção do administrador do banco de dados para a resolução da integridade dos arquivos.
- *Emergency*: o banco de dados está disponível para um único usuário, administrador do banco de dados, que precisa realizar uma manutenção com o banco ativo.

A inicialização de um banco de dados é o recurso utilizado pelo DBA para tornar o banco disponível para uso. Sem isso, os usuários não conseguem estabelecer a conexão com o banco de dados e utilizar seus recursos. A conexão com o banco de dados é definida a partir do *driver* do banco de dados, disponibilizado pelos SGBDs.

Considere a instalação de um SGBD utilizando a arquitetura **cliente-servidor**. Segundo Graves (2003), essa é uma arquitetura dividida em duas camadas. Uma camada é o servidor com o SGBD e a outra camada é o cliente, processando os dados e realizando as tarefas simples e de formação. O *driver* é uma interface, instalada no cliente, que estabelece a sua ligação com o servidor, criando uma ponte para viabilizar a transferência dos dados (Figura 1).



A desativação de um banco de dados é o recurso utilizado pelo DBA para tornar o banco indisponível para uso. A desativação desencadeia uma sequência de procedimentos que realizam uma varredura e o fechamento dos processos ativos.



### Link

O *driver* utilizado pelo SGBD MySQL é o Connector/ODBC. Todo o processo de baixa e instalação pode ser consultado na documentação oficial, disponível no *link* a seguir.

<https://qr.go.page.link/YATP>

## Inicialização de um SGBD — apresentação

A inicialização do banco de dados varia de SGBD para SGBD. Em alguns, utiliza-se a instrução `STARTUP`. Em outros, `START`. No caso do SGBD MySQL, é utilizado o conceito de servidor e, dessa forma, o servidor **mysqld**

deve ser iniciado. É importante você conhecer o local de instalação do MySQL, pois arquivos que são manipulados na ativação e na inativação estão contidos nessa estrutura da instalação.

Para a inicialização acontecer, o SGBD busca um *script* de parâmetro de inicialização do servidor, em que as configurações são definidas e ativadas. Muitas configurações podem e devem ser definidas nesse momento. As configurações são definidas para conexão e processamento.

## Configurações de conexão

A conexão é um fator importante no funcionamento de um banco de dados. Por isso, deve estar configurada de forma a não causar paradas e problemas de desempenho nem gerar indisponibilidade do banco. Segundo Fields e Kolb (2000), deve-se evitar sucessivas tentativas de conexões, pois elas representam um processo demorado e, a cada busca de informação, a conexão via *driver* é estabelecida. Estas são as configurações:

- função utilizada para abrir a comunicação do banco por meio do *driver*;
- *pool* de conexões (quantidade mínima/máxima, tempo máximo de conexões, tempo máximo de espera para conexão).

## Configurações de processamento

Estas são as configurações de processamento:

- aplicação de *case sensitive*;
- número de linhas para preenchimento de cursores, entre outras;
- opção de busca de data/hora (banco de dados ou computador cliente);
- opções de travamento no nível de colunas;
- tamanho do cache (memória) para planos com consultas frequentes;
- tamanho do cache (memória) para planos com consultas não frequentes;
- utilização de *multi-threads*.

Muitos outros parâmetros podem ser disponibilizados pelos SGBDs. Fato é que, sempre que atuar como DBA, você deverá se preocupar em criar o arquivo de configuração para que o banco de dados, objeto de sua administração, tenha um bom desempenho, um bom padrão de utilização e um risco de indisponibilidade diminuído.



No momento da inicialização, há também a opção de ativação dos arquivos de *log*, em que serão gravadas as ocorrências de execução. Esses arquivos são importantes em caso de falhas, pois servirão como fonte de rastreamento do problema, em busca de uma solução. Segundo Silberschatz, Korth e Sudarshan (2012), somente a operação executada no item de dados e o nome do item de dados são armazenados no *log*. O *log* de operação é chamado de *log* lógico.

No caso do MySQL, caso não haja o arquivo de *log* quando o servidor for iniciado, ocorrerá o *crash recovery*. Ele consiste em um processo extremamente lento de leitura de todos os arquivos lógicos do InnoDB.

Em função de análises de desempenho, ou mesmo da utilização dos usuários e da execução dos aplicativos, o DBS precisa realizar ajustes nas configurações. Quando isso ocorre e o banco de dados está “no ar”, ou seja, disponível para as conexões do usuário, deve-se verificar a necessidade de reinicializar o banco de dados. A reinicialização configura-se como uma nova forma de inicialização, pois para isso o banco de dados deve ser desativado e inicializado novamente.

## Inicialização de um SGBD — procedimentos

A seguir, você vai ver os procedimentos de inicialização do SGBD MySQL, que variam de acordo com o sistema operacional utilizado. O processo de inicialização do servidor MySQL é representado na Figura 2.



Figura 2. Processo de inicialização do MySQL.

Veja agora a descrição dos elementos mostrados na Figura 2.

- **mysql.server:** *script* que envolve o serviço MySQL. Chamado a nível de execução do sistema operacional.
- **mysqld\_safe:** processo que configura os *logs* de erro. Inicia e monitora o **mysqld** no sistema operacional.

- **mysqld**: processo que mantém o serviço em execução, depurando as funcionalidades do servidor MySQL junto ao sistema operacional, o que inclui arquivos físicos e alocação dinâmica de memória.

Este é o comando de inicialização no console:

```
'<endereço>/mysql start'
```

O formato do comando para a inicialização do banco de dados MySQL no servidor é:

```
C:\> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld"
```

O processo de inicialização verifica o arquivo de inicialização (configurações) do MySQL, inicializando o serviço automaticamente. O comando suporta os parâmetros `stop` e `reload`, para parada do servidor e recarga.

Segundo o *Manual de Referência do MySQL 8.0*, o processo de inicialização se dá por meio do console ou como serviço do servidor do sistema operacional. Veja o Quadro 1.

**Quadro 1.** Comandos de inicialização versus sistema operacional

Meio	Comando
Console	Mysqld
Windows	Serviço
Linux	Mysqld_safe mysql.server
macOS	Daemon launchd (por meio da inicialização do servidor) Daemon
Solaris	SMF ( <i>Service Management Framework</i> ) mysql.server



A seguir, veja as particularidades.

- O MySQL para Windows suporta conexões de memória compartilhada. Para isso, o servidor deve ser iniciado com a opção `--shared-memory`.
- O caminho para o **mysqld** pode variar dependendo do local de instalação do MySQL no seu sistema.
- O *log* de erros está localizado, por padrão, no diretório **C:\Program Files\MySQL\MySQL Server 8.0\data**.
- O arquivo de *log* possui um sufixo de **.err**, mas pode ser especificado na linha de comando, a partir da inclusão da opção `--log-error`.
- A inicialização do banco de dados ocorre com as opções `-standalone` e `-debug`. Um arquivo **mysqld.trace** será escrito.
- Para conhecer todas as opções que o MySQL oferece, use o comando `mysqld --verbose --help`.

## Desativação de um SGBD — apresentação

A desativação do banco de dados se dá a partir da execução da instrução **SHUTDOWN**. Essa é uma instrução que retira o banco de dados do modo *on-line*. Isso pode ser necessário por diversas razões e, dependendo da razão, o banco de dados fica em determinado estado.

A desativação de um SGBD acontece a partir da submissão do comando **SHUTDOWN** ou pelo mal funcionamento de algum serviço, seja *hardware* ou *software*. Segundo o *Manual de Referência do MySQL 8.0*, no processo de desligamento do servidor, no caso da submissão do comando **SHUTDOWN**, são disparados os seguintes procedimentos:

- desabilitação de *logons* (exceto para membros das funções de administrador de banco de dados), de modo que nenhum usuário conseguirá conexão;
- criação de um procedimento de encerramento em que serão verificados os processos em andamento — no caso do MySQL, os processos são interrompidos, mas em outros SGBDs espera-se a conclusão de procedimentos armazenados ou instruções de transação em execução;
- inserção de um ponto de verificação em cada banco de dados;
- encerramento do servidor e fechamento de mecanismos de armazenamento;
- definição do servidor como “fora do ar”.

Após o encerramento, o banco de dados assume os seguintes estados:

- 0 — finalização com sucesso (não é feito nenhum reinício);
- 1 — finalização sem sucesso (não é necessário reiniciar);
- 2 — finalização sem sucesso (reinicialização).

O processo de reinicialização, em caso de encerramento do banco de dados “com defeitos”, implica executar um processo de recuperação automática na próxima inicialização.

## Desativação de um SGBD — procedimentos

Segundo o *Manual de Referência do MySQL 8.0* (MYSQL, 2019, documento *on-line*), para o processo de desligamento do servidor, o comando de desativação no MySQL é:

```
C:\> "<endereço>\mysqladmin" -u root shutdown
```

ou

```
C:\> "<endereço>\mysqladmin" -p -u root shutdown
```

Este último é utilizado quando uma senha tiver sido atribuída ao usuário *root*. A senha será fornecida em execução. O uso da instrução *SHUTDOWN* minimiza a quantidade de trabalho de recuperação automática.



### Fique atento

Sempre que o banco de dados está no modo *shutdown*, todos os arquivos encontram-se fechados e o banco de dados não existe. Apenas quando o banco for aberto é que o usuário poderá estabelecer uma nova conexão e acessar os dados.

No estado *on-line*, o banco de dados está iniciado, montado e aberto. Nesse estado, qualquer usuário pode estabelecer conexão com o banco de dados e utilizar os comandos de acordo com a sua permissão.

## Inicialização e desativação — erros e soluções

Como você deve imaginar, a execução do banco de dados está suscetível a falhas. Segundo Alapati (2009), uma falha em um banco de dados pode provocar dados danificados, ausentes ou até perda de arquivos. Diante disso, o tratamento de uma falha é imprescindível para manter a qualidade dos dados.

Uma falha é a consequência de algum problema ocorrido na execução de uma instrução. Segundo Silberschatz, Korth e Sudarshan (2012), existe grande variedade de falhas, como quebra de disco, falha de energia, erro de *software*, fogo na sala de equipamento ou mesmo sabotagem.

O sistema de banco de dados deve definir procedimentos de prevenção para garantir as propriedades de atomicidade e durabilidade nas transações diante das falhas. Entende-se por atomicidade a capacidade de uma transação ser atualizada como um todo. A durabilidade implica que, uma vez gravadas no disco, as informações não serão apagadas e estarão disponíveis corretamente. A seguir, veja a descrição de algumas falhas.

- Falha de transação: ocorre por erro lógico quando uma condição interna causa o erro. É o caso de: entrada inadequada, dado não encontrado, *overflow* ou limite de recursos excedido e erro de sistema, como *deadlock*.
- Queda do sistema: mau funcionamento do *hardware* ou *bug* no *software* do banco de dados ou sistema operacional. Normalmente, provoca a perda do conteúdo do armazenamento volátil. O armazenamento não volátil permanece intacto. Essa condição é conhecida como “falhar-parar”.
- Falha de disco: falha na transmissão de dados ou danificação do *hardware*.

Ainda de acordo com Silberschatz, Korth e Sudarshan (2012), a recuperação das falhas está relacionada à identificação de modos de falhas e à maneira como afetam os conteúdos. Algoritmos de recuperação podem ser desenvolvidos para correção, mas são definidos em duas partes:

- ações tomadas durante o processamento normal da transação, a fim de garantir que haja informação suficiente para permitir a recuperação de falhas;
- ações tomadas em seguida à falha, recuperando o conteúdo do banco de dados para um estado que assegure a consistência, a atomicidade e a durabilidade dos dados.

Dessa forma, pode-se adotar como uma boa administração a implantação do uso de arquivos de log e a realização periódica de cópias de segurança (*backup*). A partir desses procedimentos, a recuperação será possível.

Os sistemas de banco de dados utilizam a recuperação baseada em *log*. De acordo com Silberschatz, Korth e Sudarshan (2012), o *log* é uma sequência de registros das atividades realizadas no banco de dados. Um registro de *log* identifica as atividades com: identificador de transação + identificador de item de dado + valor antigo + valor novo. A partir dessa movimentação, é possível mandar a alteração para o banco ou inutilizá-la (*undo*), caso necessário.



### Fique atento

O arquivo de *log* deve ser gravado em espaços de armazenamento estáveis, seguros.

A recuperação utiliza o procedimento de REDO, que define o valor de todos os itens de dados atualizados para novos valores. O SGBD MySQL executa o processo `mysqld _ safe`. Entre as suas funções na inicialização, está a de configurar os *logs* de erros. Algumas situações podem ocorrer:

- caso o serviço **mysqld** seja finalizado de forma anormal, a monitoração do `mysql _ safe` irá reiniciá-lo.
- caso o **mysqld** não se inicie, o *log* de erros será verificado para a análise das mensagens gravadas e a indicação da causa do problema (para a busca da solução).

No caso de situações anormais ocorridas na execução de um SGBD, mensagens de erro são apresentadas para identificar o problema. Assim, o usuário ou DBA pode buscar a solução. No site oficial dos SGBDs, você encontra a descrição das mensagens de erro codificadas para a identificação do erro e o tratamento para o desenvolvimento da solução do problema.

Cada SGBD define um padrão de escrita para a mensagem, mas eles utilizam basicamente as seguintes informações: código da mensagem + nível de severidade + existência de *log* + texto explicativo. No SGBD MySQL, a mensagem de erro se apresenta no seguinte formato:

```
shell> mysql -h no-such-host
```

```
ERROR 2005 (HY000): Unknown MySQL server host 'no-such-host' (0)
```

Na codificação da mensagem, os erros normalmente são classificados em:

- informativos;
- de *software*;
- de mecanismos de banco de dados (armazenamento, segurança e processamento).

Em sua maioria, as falhas que geram mensagens informativas não geram erro. As falhas relacionadas ao *software* são facilmente corrigidas pelo próprio usuário. Contudo, as falhas geradas em relação ao armazenamento, à segurança e ao processamento do SGBD devem ser analisadas e reparadas pelo DBA, pois podem gerar danos nos dados, provocando atitudes de correção. Você pode tentar iniciar o servidor com a opção `--console`. Nesse caso, o servidor apresenta algumas informações na tela para auxiliar na identificação do problema.

As mensagens de erro, por padrão, são em inglês, mas o SGBD MySQL disponibiliza um recurso para que elas sejam exibidas ao usuário em outro idioma, como: checo, dinamarquês, holandês, estoniano, francês, alemão, grego, húngaro, italiano, japonês, coreano, norueguês, polonês, português, romeno, russo, eslovaco, espanhol ou sueco. Esse efeito afeta também as mensagens gravadas no *log* de erros. O comando MySQL para configurar o idioma desejado é:

```
mysqld --lc_messages_dir=/usr/share/mysql  
--lc_messages=localidade
```



### Link

O **mysqld** mapeia para a localidade e o idioma correspondente. Deve existir um arquivo com o nome do idioma no diretório **/usr/share/mysql/idioma**, em que serão gravadas as informações de *log*. Você pode ver mais detalhes no *link* a seguir.

<https://qrgo.page.link/BNG9>

O MySQL classifica os erros em três categorias: servidor, cliente e geral. Entre os erros mais comuns no SGBD MySQL, estão:

- a senha falha quando inserida interativamente;
- a tabela **nome\_tabela** não existe;
- o acesso é negado;
- o arquivo não é encontrado e erros semelhantes;
- os comandos estão fora de sincronia;
- a conexão com o servidor **mysql** é perdida;
- o disco está cheio ou a tabela está no limite de espaço;
- a comunicação é problemática e há conexões abortadas;
- há excesso de conexões;
- o host **host\_name** está bloqueado;
- o usuário é ignorado;
- não é possível conectar-se ao servidor **mysql** (local);
- não é possível criar/gravar em arquivo;
- não é possível inicializar o conjunto de caracteres;
- o cliente não suporta o protocolo de autenticação;
- o pacote é muito grande;
- há problemas de corrupção de tabelas;
- o servidor **mysql** foi embora.



### Link

No *link* a seguir, você pode ver o detalhamento de cada erro.

<https://qrgo.page.link/E6dD>

Segundo o Manual de Referência do MySQL 8.0/Manipulação de Erros do InnoDB, a definição de parâmetros na criação do banco de dados influencia em seu funcionamento, pois em caso de esgotamento de espaço, *deadlock* (esgotamento do tempo de espera), detecção de chave duplicada ou tamanho grande de registro na gravação ocasiona o *rollback* da instrução, gerando baixa performance ou ainda travamento, levando à desativação.



Muitos desses erros podem ser evitados caso sejam definidos os parâmetros de configuração de inicialização do banco de dados a partir de uma análise de comportamento do ambiente. Tal análise levaria em conta a frequência de acesso, o volume de dados e o tipo de operações frequentes.

Analisando os erros elencados, note a importância da configuração definida para a inicialização do banco de dados. Ela influencia diretamente o processamento, podendo gerar erros ou apresentar procedimentos diferentes de recuperação. Dessa forma, é importante que, antes de planejar a configuração das opções de inicialização, você:

- conheça os parâmetros de inicialização e a arquitetura do SGBD;
- conheça a necessidade real do ambiente;
- configure apropriadamente o arquivo de *log*.



## Referências

ALAPATI, S. *OCF Oracle Database 11g: novos recursos para administradores: guia do exame 1Z0-050*. Porto Alegre: Bookman, 2009.

FIELDS, D. K.; KOLB, M. A. *Desenvolvendo na Web com JavaServer Pages*. Rio de Janeiro: Ciência Moderna, 2000.

GRAVES, M. *Projeto de banco de dados com XML*. São Paulo: Pearson, 2003.

MYSQL. *Chapter 4 Connector/ODBC Installation*. 2019. Disponível em: <https://dev.mysql.com/doc/connector-odbc/en/connector-odbc-installation.html>. Acesso em: 14 abr. 2019.

MYSQL. *Common Errors When Using MySQL Programs*. 2019. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/common-errors.html>. Acesso em: 14 abr. 2019.

MYSQL. *Download MySQL Installer*. 2019. Disponível em: <https://dev.mysql.com/downloads/windows/installer/8.2.html>. Acesso em: 14 abr. 2019.

MYSQL. *Iniciando e Parando o MySQL Automaticamente*. 2019. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/automatic-start.html>. Acesso em: 14 abr. 2019.

MYSQL. *Manipulação de Erros do InnoDB*. 2019. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/innodb-error-handling.html>. Acesso em: 14 abr. 2019.

MYSQL. *O processo de desligamento do servidor*. 2019. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/server-shutdown.html>. Acesso em: 14 abr. 2019.

MYSQL. *Setting the Error Message Language*. 2019. Disponível em: <https://dev.mysql.com/doc/refman/5.5/en/error-message-language.html>. Acesso em: 14 abr. 2019.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de banco de dados*. Rio de Janeiro: Elsevier, 2012.

## Leituras recomendadas

MANNINO, M. V. *Projeto, desenvolvimento de aplicações e administração de banco de dados*. 3. ed. Porto Alegre: McGraw-Hill, 2008.

MSFT, M. *et al*. Severidade dos erros do Mecanismo de Banco de Dados. *Microsoft*, 2016. Disponível em: <https://docs.microsoft.com/pt-br/sql/relational-databases/errors-events/database-engine-error-severities?view=sql-server-2017>. Acesso em: 14 abr. 2019.

RAY, M. *et al*. Iniciar, parar, pausar, retomar e reiniciar os serviços SQL Server. *Microsoft*, 2016. Disponível em: <https://docs.microsoft.com/pt-br/sql/database-engine/configure-windows/start-stop-pause-resume-restart-sql-server-services?view=sql-server-2017>. Acesso em: 14 abr. 2019.

STEIN, S. *et al*. Estados de banco de dados. *Microsoft*, 2016. Disponível em: <https://docs.microsoft.com/pt-br/sql/relational-databases/databases/database-states?view=sql-server-2017#database-state-definitions>. Acesso em: 14 abr. 2019.

TONSIG, S. L. *MySQL: aprendendo na prática*. Rio de Janeiro: Ciência Moderna, 2006.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

