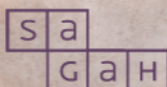


ADMINISTRAÇÃO DE BANCO DE DADOS

Maurício de Oliveira Saraiva



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Recuperação de banco de dados

Objetivos de aprendizagem

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

- Explicar o que é a restauração de um *backup*.
- Determinar a necessidade de uso de *backups*.
- Desenvolver a restauração de *backups*.

Introdução

A competição acirrada no mercado de trabalho tem feito com que as empresas invistam cada vez mais em tecnologia, especialmente para o tratamento de informações e a disponibilidade de dados, tanto para clientes internos como para clientes externos à organização. No cenário atual, os *backups* de banco de dados podem atuar como uma garantia de que as informações estarão salvas e disponíveis, afinal, eles são recursos que podem restaurá-las quando necessário, recuperando dados possivelmente perdidos.

Neste capítulo, você vai estudar os conceitos relacionados à restauração de cópias de segurança e verificar a necessidade de realizar *backups* em bancos de dados. Além disso, vai ver como ocorre a aplicação da restauração por linhas de comando no MySQL.

O que é restauração de *backup*?

O principal objetivo de implementar rotinas de *backup* de bancos de dados é justamente criar cópias de segurança que poderão ser restauradas mediante a ocorrência de determinados eventos. Esses eventos representam todos os fatores que envolvem um banco de dados, como falha de rede de comunicação, defeitos em equipamentos, atualização de *softwares*, operações executadas por pessoas, mudança de processos de negócio na organização, etc.

Bryla e Loney (2005) explicam que a recuperação de *backup* de um banco de dados consiste em restaurar a estrutura e os dados ao estado em que eles estavam no momento em que uma cópia de segurança foi realizada com sucesso. Isso significa que é possível realizar diversos *backups* de um banco de dados, considerados estados, e que a recuperação pode ser aplicada no estado desejado, levando em consideração a finalidade de determinada situação. Na Figura 1, você pode ver a dinâmica da cópia e da restauração de um *backup*.

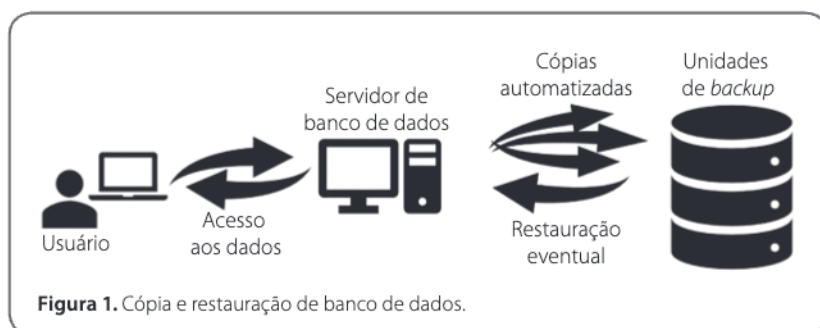


Figura 1. Cópia e restauração de banco de dados.

Conforme ilustrado na Figura 1, o usuário acessa os dados armazenados em um servidor de bancos de dados por meio de sistemas específicos. Cópias de segurança são realizadas automaticamente e armazenadas em unidades de *backup*. Quando necessário, restaurações eventuais podem ser realizadas, substituindo os dados dos *backups* de determinado estado no servidor de bancos de dados. Com isso, os usuários passam a acessar os dados atualizados.

As unidades de *backup* podem armazenar diversas cópias de segurança de um mesmo banco de dados. Isso significa que é possível escolher qual estado do banco de dados restaurar, levando em consideração os interesses do usuário gestor da informação. A restauração de um banco de dados é uma rotina de responsabilidade do administrador de banco de dados (*DataBase Administrator* [DBA]) (RAMAKRISHNAN; GEHRKE, 2011). Normalmente, usuários não possuem acesso às ferramentas de *backup* e restauração, de modo que apenas os profissionais da área de suporte de tecnologia das organizações ficam responsáveis por essas atividades.

Assim como a execução de um *backup*, a restauração é uma atividade criteriosa que deve ser realizada com bastante atenção. O profissional responsável pela recuperação de um *backup* deve seguir os passos previstos na

documentação, tomando cuidado, por exemplo, para não restaurar e substituir um banco de dados diferente do solicitado.



Fique atento

Se é possível perder informações por meio de eventos adversos que fogem ao controle dos administradores de bancos de dados, uma recuperação malsucedida também pode ocorrer, tornando os dados indisponíveis após uma tentativa frustrada de restauração (DATE, 2004).

Tipos de restauração de *backup*

A restauração de *backups* de bancos de dados tem relação com o tipo de cópia de segurança que foi realizada e com o local em que os *backups* foram armazenados. Essas informações devem estar registradas em um documento que descreve a estratégia de *backup* utilizada pela organização. Desse modo, a empresa pode definir uma forma diferente de realizar *backups* e/ou restaurações de seus bancos de dados conforme as características de seus sistemas. Para isso, pode levar em consideração diversos fatores, como a importância e a disponibilidade das informações armazenadas.

Alguns *backups* podem ser recuperados com o banco de dados no estado *on-line*, pois podem atualizar apenas algumas entidades sem restringir o acesso aos usuários, como na atualização de determinadas tabelas. Entretanto, um *backup* completo requer que o banco de dados esteja *off-line* para que seja possível substituir todos os arquivos necessários. A seguir, você vai conhecer os métodos de restauração de *backup* mais utilizados no MySQL (MYSQL, 2019a).

Restauração de *backups* físicos

A execução de *backups* físicos ou brutos é realizada pela cópia de arquivos e pastas dos bancos de dados a partir da sua localização física. Normalmente, esse tipo de cópia é realizada em grandes bancos de dados ou quando os bancos de dados precisam ser restaurados rapidamente.

A restauração de *backups* físicos é simples e envolve a descompactação (se for o caso) e a substituição dos arquivos no servidor do banco de dados.

Independentemente de a origem dos dados estar em um disco rígido em outro servidor, em unidades de fita ou até mesmo na nuvem, a substituição se dá por instruções de cópia de arquivos do próprio sistema operacional.

Alguns sistemas gerenciadores de bancos de dados armazenam os registros das tabelas em arquivos separados. Isso permite que os *backups* físicos sejam restaurados em parte. Caso contrário, apenas o banco de dados inteiro pode ser recuperado.

Restauração de *backups* lógicos

Um *backup* lógico é uma cópia de segurança que salva a estrutura e os dados armazenados em um banco de dados em arquivos que registram instruções em formato SQL. A restauração de um *backup* lógico se dá pelo processo reverso, isto é, indica-se o banco de dados que será restaurado e o arquivo que contém as instruções que foram salvas daquele banco de dados.

A restauração de um *backup* lógico também permite recuperar dados parciais em vez de todo o banco de dados. Com isso, é possível restaurar os registros de determinadas tabelas ou apenas a estrutura de elementos do banco de dados.

A recuperação completa de *backups* lógicos pode ser mais demorada em relação à restauração de cópias de segurança de *backups* físicos. Isso ocorre porque as instruções SQL são executadas em sequência para remontar a estrutura das tabelas e reinserir os registros que foram resguardados.

Restauração de *backups* incrementais (*point-in-time*)

Os *backups* incrementais utilizam os arquivos de *log* de transações dos bancos de dados. Esse método de *backup* atua em conjunto com os *backups* completos, que, devido ao seu tamanho, são executados em intervalos maiores de tempo. Nesse método, a restauração do banco de dados se dá pela recuperação do último *backup* completo, adicionando-se, em ordem cronológica crescente, os *backups* incrementais que foram executados posteriormente por meio dos arquivos de *log* de transações.

Esses arquivos de *log* contêm as operações que foram executadas nos bancos de dados, como modificações nas estruturas das tabelas efetuadas por desenvolvedores e nos registros de dados — inclusões, alterações e exclusões realizadas pelos usuários no uso dos sistemas.

O método de utilização de *backups* incrementais permite restaurar os bancos de dados em estados diferentes com pouco custo de armazenamento, pois os arquivos de *log* de transações contêm dados de pequenos intervalos de tempo, como um dia, um turno ou uma hora.

Necessidade de *backups*

Nos dias atuais, inúmeras empresas utilizam bancos de dados para armazenar informações sobre o seu negócio. Esses bancos de dados, em conjunto com sistemas informatizados, permitem processar, armazenar e disponibilizar informações de modo mais ágil e confiável. Assim, proporcionam maior produtividade e retorno dos investimentos (MICROSOFT, 2018).

Entre as vantagens do uso de bancos de dados, destacam-se as listadas a seguir.

- Armazenamento de dados: empresas armazenam em bancos de dados informações de suas operações, como registros de dados de clientes, fornecedores, faturamento, vendas, compras, pedidos, estoques, etc.
- Disponibilidade de dados: os dados armazenados precisam estar íntegros e disponíveis para usuários internos e clientes externos.
- Agilidade de processamento: sistemas computacionais conseguem processar inúmeras transações em um curto espaço de tempo, agilizando processos e promovendo maior capacidade de produção de informações.

Dessa forma, considera-se inviável a perda, a corrupção ou a indisponibilidade de dados, uma vez que isso pode promover a perda de negócios e até uma possível falência em casos mais extremos. Assim, para garantir que os dados sejam resguardados, existem as rotinas de *backup*.

A execução de *backups* de um banco de dados deve ser entendida como a garantia de que uma cópia de segurança poderá ser restaurada futuramente — quando for necessário. Para isso, é preciso considerar diversos fatores, como os recursos disponíveis, a importância da disponibilidade das informações, o conhecimento técnico da equipe responsável pela execução, etc.

Trabalhar com a execução de *backups* de bancos de dados é uma atividade criteriosa, que requer planejamento de ações e cuidado na execução. O planejamento é necessário para definir as estratégias de *backup* e restauração,

que são movidas de acordo com os objetivos e interesses da organização. Já o cuidado implica atenção especial na execução, pois um procedimento realizado de modo indevido pode tornar os dados inacessíveis definitivamente.



Fique atento

Ao definir uma estratégia de *backups*, você deve levar em consideração que ela precisa ser avaliada para ser aprovada. Para isso, é preciso realizar diversos testes de cópia e restauração, visando a garantir que um *backup* seja restaurado com sucesso quando for necessário.

A importância de fazer *backups* de bancos de dados

Tão importante quanto armazenar os dados é conseguir acessá-los e disponibilizá-los conforme os interesses da organização. Por isso, realizar cópias de segurança é uma atividade imprescindível para quem utiliza bancos de dados, pois sistemas e equipamentos estão sujeitos a eventos que podem causar falhas e impedir o acesso às informações. A seguir, você vai ver alguns dos principais eventos que podem criar a necessidade de realizar cópias de segurança de bancos de dados.

Desastres naturais e sinistros

Desastres naturais e sinistros, como terremotos, alagamentos, explosões, furtos e roubos, são eventos fortuitos que podem danificar ou cortar a comunicação com equipamentos e servidores. Por vezes, a recuperação é inviável e os dados armazenados são perdidos.

Falhas em servidores

Servidores, tanto físicos como lógicos (máquinas virtuais), podem sofrer falhas por desgaste natural, mau uso, má conservação, falta de manutenção, erros de configuração e erros de operação, entre outros, tornando os sistemas e os dados indisponíveis.

Falhas de *hardware*

Falhas de *hardware* indicam o mau funcionamento de equipamentos e dispositivos. Discos rígidos de armazenamento de dados (*Hard Disk* [HD]), por exemplo, podem sofrer danos em áreas específicas. Esses equipamentos, por serem dispositivos físicos, estão sujeitos a falhas devido a desgaste, defeito ou vício de fabricação, corrompendo os dados ali armazenados.

Exclusão acidental de dados

A exclusão de dados pode ocorrer acidentalmente por meio de instruções definidas pelos programas. Tais instruções podem ser executadas por usuários por meio da interface dos sistemas. Elas também podem ser executadas por analistas desenvolvedores e administradores de bancos de dados que executam *scripts* diretamente nos bancos de dados. Nesses casos, normalmente registros ou tabelas inteiras são excluídas, proporcionando perda involuntária de informações.

Trocas de equipamentos

As trocas de equipamentos normalmente são realizadas por defeitos, vícios ou obsolescência. Uma maneira de transportar os bancos de dados de um equipamento para outro é por meio da execução de *backup* no equipamento de origem e de sua respectiva restauração no equipamento de destino. Dessa forma, os dados são copiados com facilidade, integridade e segurança.

Atualização de sistemas

Por mais simples que pareça, a atualização de sistemas deve ser considerada uma atividade crítica. Uma boa estratégia é realizar *backups* antes de executá-la. Aplicar *patches* de segurança em sistemas operacionais ou atualizar programas gerenciadores de bancos de dados para versões mais recentes pode falhar e tornar os dados inacessíveis.

Espelhamento de servidores

Sistemas críticos que precisam de alta disponibilidade requerem que os dados sejam copiados dos servidores principais (*master*) para equipamentos secundários (*slave*) em intervalos regulares. Para isso, *backups* de *logs* de transações de bancos de dados, que contêm apenas os últimos dados modificados, são transportados para manter os servidores com dados em redundância. No caso de alguma falha no servidor principal, o servidor secundário pode assumir o controle de forma transparente para os usuários.

Restauração de *backups* por linhas de comando

Para você ver como funciona a restauração de um *backup* por linhas de comando, considere instruções do banco de dados MySQL (MYSQL, 2019b). Considere, também, o uso do banco de dados MariaDB, que possui compatibilidade e compartilha das mesmas instruções do MySQL.



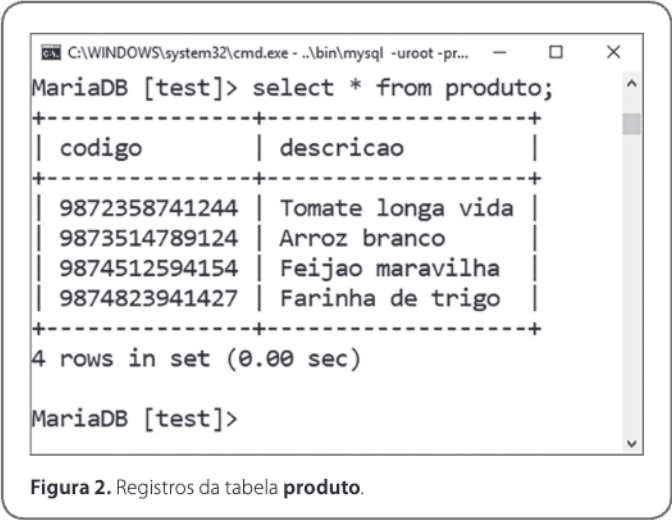
Link

Acesse os *links* a seguir para fazer o *download* do MySQL e do MariaDB, respectivamente.

<https://www.mysql.com>

<https://mariadb.org>

Para entender como funciona a recuperação de *backups* por linhas de comando no MySQL, considere um banco de dados chamado **test** que contém uma tabela chamada **produto**. Inicialmente, essa tabela possui quatro registros, como você pode ver na Figura 2.



```
C:\WINDOWS\system32\cmd.exe - ..\bin\mysql -uroot -pr...
MariaDB [test]> select * from produto;
+-----+-----+
| codigo      | descricao      |
+-----+-----+
| 9872358741244 | Tomate longa vida |
| 9873514789124 | Arroz branco    |
| 9874512594154 | Feijao maravilha |
| 9874823941427 | Farinha de trigo |
+-----+-----+
4 rows in set (0.00 sec)

MariaDB [test]>
```

Figura 2. Registros da tabela **produto**.

Para cada tipo de restauração de *backups* apresentado a seguir, considere os seguintes passos:

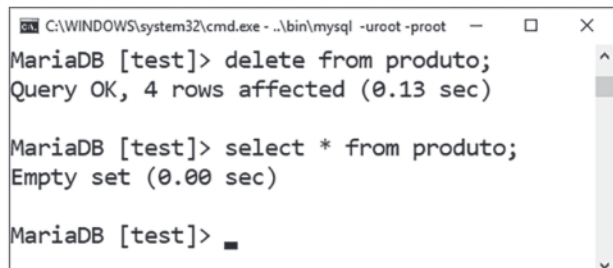
- fazer um *backup* para salvar uma cópia do banco de dados **test**;
- excluir os registros da tabela **produto**;
- exibir a tabela **produto** sem registros para comprovar a exclusão;
- restaurar o banco de dados **test**;
- exibir os dados da tabela **produto** para demonstrar a recuperação dos registros.

Restauração de *backup* lógico

Para realizar o *backup* lógico do banco de dados **test**, é preciso executar a seguinte instrução no *prompt* de comando. Essa instrução utiliza o usuário **root**, a senha **root** e produz um arquivo chamado **test.sql**, que contém a estrutura e os dados do respectivo banco de dados. Veja:

```
mysqldump -uroot -proot test > test.sql
```

Na sequência, é preciso efetuar *login* no MySQL, excluir os registros da tabela **produto** e listar o conteúdo da tabela para mostrar que não há registros gravados (Figura 3).



```
C:\WINDOWS\system32\cmd.exe - ..\bin\mysql -uroot -proot
MariaDB [test]> delete from produto;
Query OK, 4 rows affected (0.13 sec)

MariaDB [test]> select * from produto;
Empty set (0.00 sec)

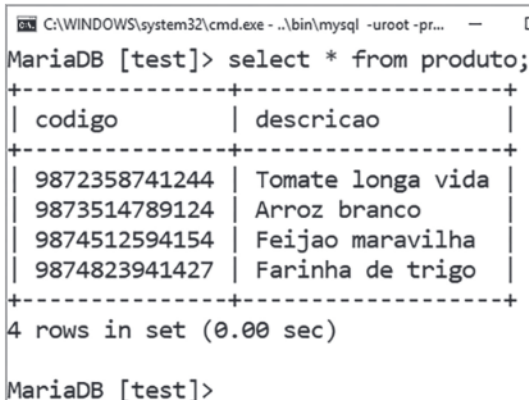
MariaDB [test]> █
```

Figura 3. Exclusão dos registros da tabela **produto**.

Por fim, é necessário restaurar o *backup* lógico que foi realizado para recuperar os registros excluídos da tabela **produto**. No *prompt* de comando, entre com a seguinte instrução:

```
mysql -uroot -proot test < test.sql
```

Nessa instrução, você utiliza o usuário `root`, a senha `root`, o banco de dados `test` e o arquivo de *backup* `test.sql`. O resultado realiza a recuperação dos registros que haviam sido excluídos da tabela **produto**, como você pode ver na Figura 4.



```
C:\WINDOWS\system32\cmd.exe - ..\bin\mysql -uroot -pr...
MariaDB [test]> select * from produto;
+-----+-----+
| codigo      | descricao      |
+-----+-----+
| 9872358741244 | Tomate longa vida |
| 9873514789124 | Arroz branco    |
| 9874512594154 | Feijao maravilha |
| 9874823941427 | Farinha de trigo |
+-----+-----+
4 rows in set (0.00 sec)

MariaDB [test]>
```

Figura 4. Registros restaurados da tabela **produto**.

Restauração de *backup* completo e incremental (*point-in-time*)

A forma mais simplificada de executar *backups* físicos completos do MySQL é a cópia e/ou compactação dos arquivos de banco de dados da pasta **mysql/data/nome_banco_de_dados** para outro local, que pode ser um servidor, uma unidade de fita ou um local na nuvem.

É preciso parar o serviço MySQL para realizar a cópia dos arquivos por comandos do sistema operacional. Caso contrário, podem surgir erros de violação de compartilhamento. Uma vez que o serviço MySQL é parado, o arquivo de *log* de registro binário de transações atual é fechado. Um novo será aberto quando o serviço for iniciado.

Nesse caso, inclua novamente os quatro registros na tabela **produto** e finalize o serviço MySQL para forçar a criação de um novo arquivo de *log* binário contendo os registros incluídos. Na sequência, exclua novamente esses registros da tabela **produto** e finalize outra vez o MySQL para fazer com que um novo arquivo de *log* binário seja criado, como ilustra a Figura 5, a seguir.

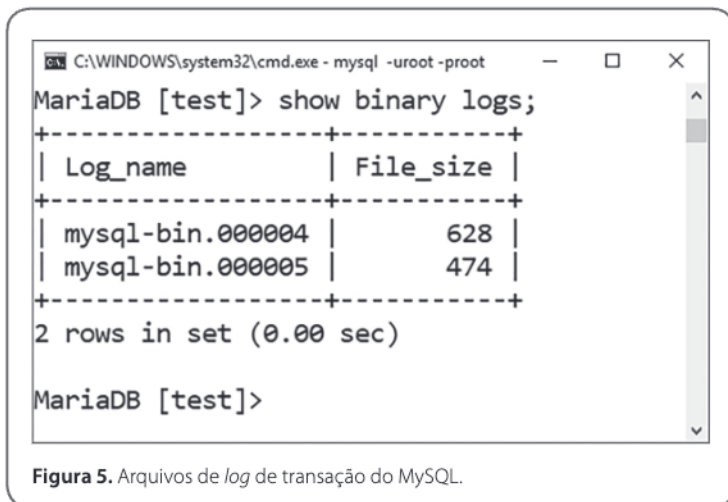


Figura 5. Arquivos de *log* de transação do MySQL.

Em relação aos arquivos de *log* binário apresentados, sabe-se que o primeiro (**mysql-bin.000004**) contém os registros incluídos na tabela **produto** e que o segundo (**mysql-bin_000005**) contém os comandos de exclusão deles. Dessa forma, você deve entrar com a seguinte instrução para reverter o banco de dados **test** ao estado em que os registros estavam na tabela **produto**:

```
mysqlbinlog mysql-bin.000004 | mysql -u root -proot  
test
```

Essa instrução reverte o banco de dados ao estado em que se encontrava quando o arquivo de *log* binário **mysql-bin.000004** foi encerrado, eliminando as operações de exclusão de registros que foram realizadas após e que estão registradas no arquivo **mysql-bin.000005**.



Referências

- BRYLA, B.; LONEY, K. *Oracle 10g: o manual do DBA*. Rio de Janeiro: Elsevier, 2005.
- DATE, C. J. *Introdução a sistemas de bancos de dados*. São Paulo: Elsevier, 2004.
- MYSQL. *Database backup methods*. 2019a. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/backup-methods.html>. Acesso em: 15 abr. 2019.
- MYSQL. *Backup and recovery*. 2019b. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html>. Acesso em: 15 abr. 2019.
- RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: Penso, 2013.

Leituras recomendadas

- MARIADB FOUNDATION. 2019. Disponível em: <https://mariadb.org/>. Acesso em: 15 abr. 2019.
- RAY, M. Fazer backup e restaurar bancos de dados do SQL Server. *Microsoft*, 2018. Disponível em: <https://docs.microsoft.com/pt-br/sql/relational-databases/backup-restore/backup-and-restore-of-sql-server-databases?view=sql-server-2017>. Acesso em: 15 abr. 2019.

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

Conteúdo:

