



Desenvolvimento de Sistemas

Recuperação de dados: *backup*, *restore*, planos para desastres

Contexto

É importante que você tenha *backups* regulares de seus bancos de dados para que, caso haja algum problema, você consiga restaurar e manter executando os seus dados.

Muitos problemas podem causar a perda de dados de um banco de dados, como quedas de sistemas (*crashes*), erros de *hardware*, apagamento de dados por usuários e/ou erros desses dados. *Backups* também são essenciais como uma medida de segurança para uma eventual transferência de dados ou até mesmo *upgrade* de *hardware* ou de *software*.

Planos para desastres

Toda a empresa que confia na tecnologia da informação e/ou a utiliza para manter seus dados e fazer a manutenção deles está sujeita a um desastre de perda de dados e também de falha de equipamentos. É necessário, então, um plano de ação para tais desastres, para que, com efetividade e rapidez, os dados sejam novamente colocados *on-line*.



Figura 1 – Servidor após uma falha no sistema de ar-condicionado que resultou em um incêndio

Fonte: <<http://www.titanpower.com/blog/wp-content/uploads/2017/04/server-room-fire.jpg>>. Acesso em: 17 mar. 2022.

A figura mostra, ao centro, um bombeiro andando dentro de uma sala que foi destruída pelas chamas. À direita estão racks de servidores queimados e caídos.

Um plano para desastres não tem apenas como função recuperar dados perdidos e restaurar os sistemas, mas também ajudar a criar métodos de detecção e prevenção de problemas.

Qualquer pessoa responsável por criar e aplicar esses planos precisa ser um pessimista com uma boa imaginação para que idealize diversos cenários possíveis.

Nem sempre, porém, o desastre é completo, e o *backup* serve para restaurar dados também no caso de vários outros tipos de problemas. Servidores contam com peças de *hardware*, as quais são suscetíveis a falhas e erros. As pessoas também

estão sujeitas a errar e a apagar dados que não deveriam ser apagados.



Falhas na alimentação de energia, eletricidade estática e até mesmo pequenos tremores de terra podem causar problemas nos dados, que podem ser contornados, caso os *backups* estejam atualizados.

Mas, afinal, o que são *backup* e restauração de dados?

Backup

É uma prática que consiste em **salvar cópia dos dados do seu servidor em um ou mais lugares além do próprio servidor**, para que, em uma eventual perda de dados, estes possam ser restaurados sem maiores problemas.

Restauração

A restauração de dados, como o próprio nome diz, consiste nas práticas de **recuperar dados originais por meio de um arquivo de *backup***.

Ao se preparar um plano de desastre de uma empresa, é sempre bom considerar o pior dos cenários, tendo como ponto de partida a hipótese de que todos os dados podem ter sido perdidos ou comprometidos. Após uma pandemia, uma revolução ou um tsunâmi, por exemplo, como os dados poderiam ser restaurados?

Porém, um plano de recuperação de dados de uma empresa é apenas parte do trabalho. O ideal é que o plano seja executado diariamente em todas as tarefas que envolvem dados, para assegurar uma maior segurança e a prevenção de problemas.

Para planejar adequadamente a restauração de dados de uma empresa, é preciso considerar um contexto bastante amplo, não sendo suficiente apenas criar e manter *backups*, pois isso implica que o prédio em que estes estão armazenados

ainda esteja em pé. Além disso, deve-se ponderar, por exemplo, se, ao restaurar um *backup* muito grande, a rede suportaria o tráfego ou se o *hardware* suportaria o processamento desses dados. Pensar em todas essas possibilidades é o que torna um plano de desastres realmente sólido e conciso.

Plano de recuperação de dados

Todo plano de recuperação de dados precisa ser o mais simples possível, pois, dado o grande espectro de desastres, quanto mais simples de ser implementado, melhor o plano será. É importante que o plano considere as prioridades de serviços que devem ser recuperados. Por exemplo, é imprescindível que, no caso de falha de um sistema bancário, as informações de contas e usuários sejam os primeiros serviços a voltarem a funcionar.

A ideia de preparar e escrever planos para restaurar bancos de dados e instalar sistemas pode parecer um trabalho incômodo e maçante. Se você perguntar a administradores de sistemas sobre os planos deles de recuperação de dados, é possível que ouçam apenas algumas palavras e comandos que seriam executados.

Na realidade, porém, quando um plano desses é necessário – em alguma situação de desastre –, todos os procedimentos planejados e estruturados serão fundamentais para uma restauração dos dados com segurança e velocidade.

Reforce-se que o termo “desastre” utilizado neste contexto não implica necessariamente em situações de perigo ou destruição. Um desastre pode ser, por exemplo, um programador que, sem querer, apaga todos os registros de uma tabela em um banco de dados de produção (ou do cliente), ou seja, um fato de grande trivialidade, mas que causa muitos problemas.

Um bom inventário de todos os dados de uma empresa é imprescindível. Para recuperar dados após uma falha total, é preciso saber onde os dados estão armazenados. Também é necessário entender que “dados” são as informações

armazenadas em computadores, servidores, banco de dados, *laptops*, envelopes e inclusive na memória de funcionários, como *passwords*, IDs, chaves de criptografia etc. Tudo isso deve ser levado em conta na hora de planejar como recuperar esses dados.

Backups

O MySQL e o Workbench Community já vêm com um serviço de *backup* que executa automaticamente à meia-noite. Caso você esteja utilizando seu computador nesse horário, poderá ver em sua tela uma janela de comando com algumas informações do *backup* que está sendo executado. Esse recurso, no entanto, pode ser desativado e nem sempre se pode contar com isso.

É necessário compreender, antes de qualquer coisa, como realizar um *backup* no momento em que ele for necessário. Para efetuar o *backup* manual de todo o banco de dados no Workbench, faça o seguinte:

- ◆ Conecte-se ao banco de dados com o Workbench.
- ◆ Clique em **Server** na barra de ferramentas principal.
- ◆ Selecione **Data Export**.
- ◆ Selecione as tabelas das quais deseja fazer o *backup*.
- ◆ Em **Export Options**, selecione onde você pretende salvar o *backup* (*dump*). Por padrão, o salvamento fica na pasta **Documents** em uma subpasta chamada **dumps**.

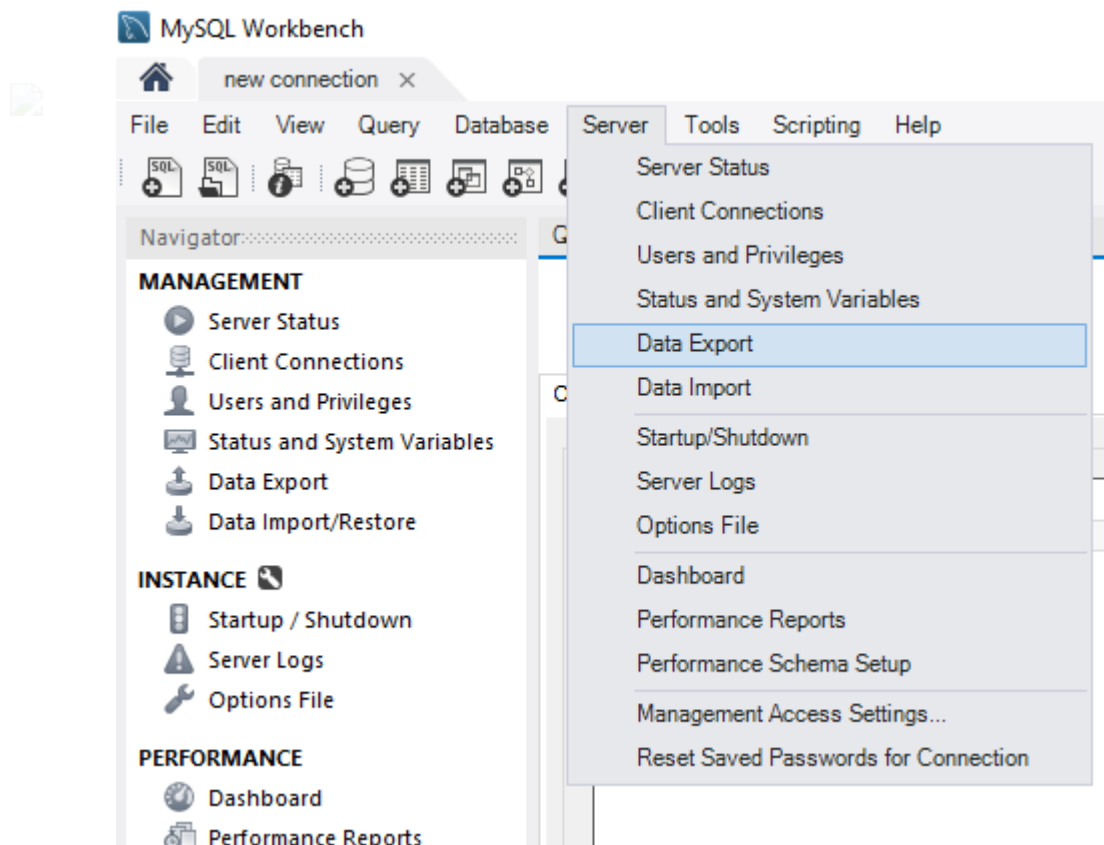


Figura 2 – Selecionando o **Data Export**

A figura mostra a janela do programa MySQL Workbench com menu “Server” selecionado e a opção “Data Export” selecionada.

Na aba que abre de Data Export temos algumas opções bastante importantes que temos que levar em conta.

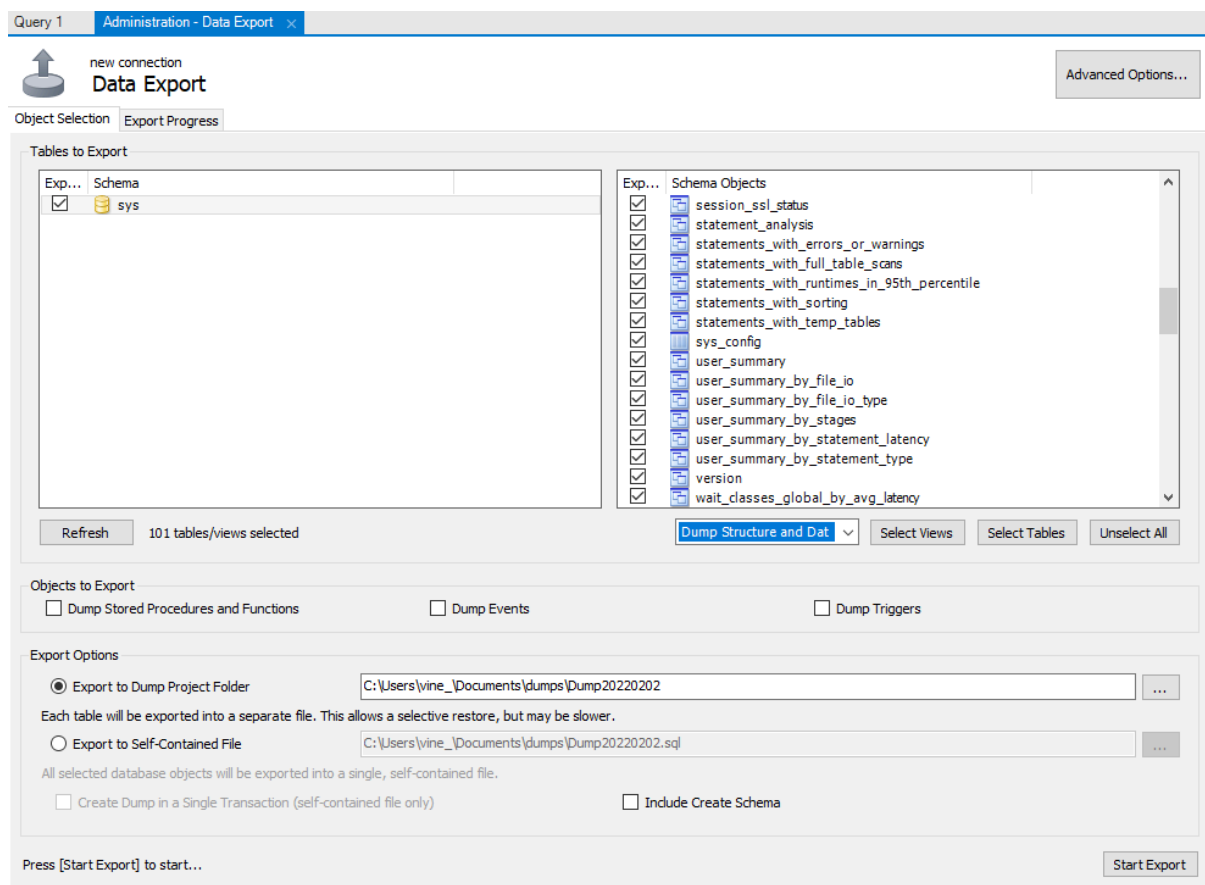


Figura 3 – Preparando a exportação do *backup*

A figura mostra a janela do programa MySQL Workbench após selecionada a opção “Data Export”, o que foi representado na imagem 2. Nessa janela estão, à esquerda, uma caixa de seleção na qual se pode escolher o banco do qual se quer fazer o backup e, à direita, outras caixas de seleção para se escolher de que tabelas do banco escolhido os backups serão gerados. Abaixo e à direita está o botão “Start Export”.

A figura mostrou a opção de exportar cada tabela em um arquivo separado (**Export to Dump Project Folder**), o que possibilita, em caso de perda de dados, escolher exatamente quais tabelas serão restauradas e quais não se quer restaurar.

Caso fosse selecionada a opção de exportação para um só arquivo (**Export to Self-Contained File**), a exportação seria feita para apenas um arquivo com todos os dados selecionados para a exportação.

Atente-se que é possível exportar também eventos, *triggers* e funções do banco de dados, selecionando-os em **Objects to Export**.

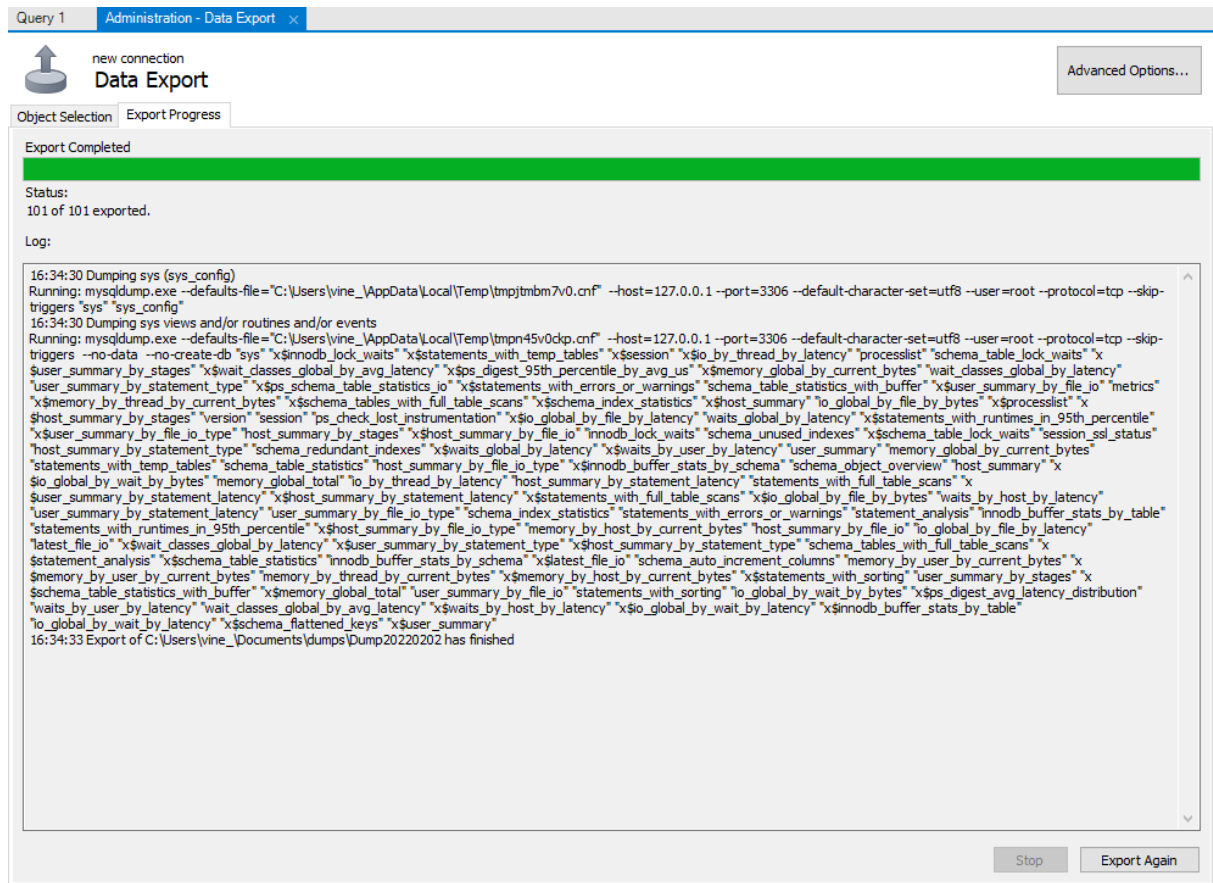


Figura 4 – Janela de *log* do *backup* sendo executada

A figura mostra uma janela do programa MySQL Workbench após o botão “Start Backup” da imagem 3 ser pressionado. Nessa janela estão, ao centro, todos os comandos automáticos gerados pelo algoritmo que está executando o backup dos bancos (*log*). Abaixo e à direita está um botão chamado “Export Again”.

Observe que o *backup* começa a ser feito na pasta temporária e, ao término, é copiado para a pasta **Documents**.

Após efetuado o *backup*, os arquivos de *backup* são criados e armazenados

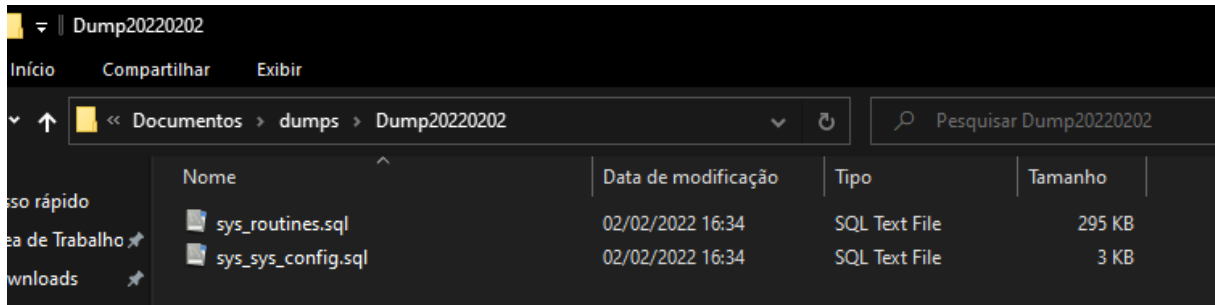


Figura 5 – Arquivos de *backup* criados e armazenados

A figura mostra o explorador de arquivos do Windows na pasta “Documentos/dumps/Dump20220202”, com dois arquivos com extensão “.sql” dentro dessa pasta. Um arquivo tem o nome de “sys_routines.sql” e o outro tem o nome de “sys_sys_config.sql”.

Quando o *backup* for concluído, os arquivos SQL do banco de dados estarão salvos na pasta **dumps**.

Restore (restauração de um *backup*)

Antes de efetuar qualquer restauração, é preciso tome cuidado com dados que tenham sido inseridos após a perda dos dados e antes da restauração, pois eles podem acabar sendo sobrescritos.

Com o MySQL Workbench 8, é possível restaurar um *backup* de maneira muito simples. Após conectar-se ao servidor, na área à esquerda, em **Navigador**, use a aba **Administration**, que está disponível no rodapé dessa área. Nela, há as opções de *status*, conexões, importação e exportação de dados, entre outras.

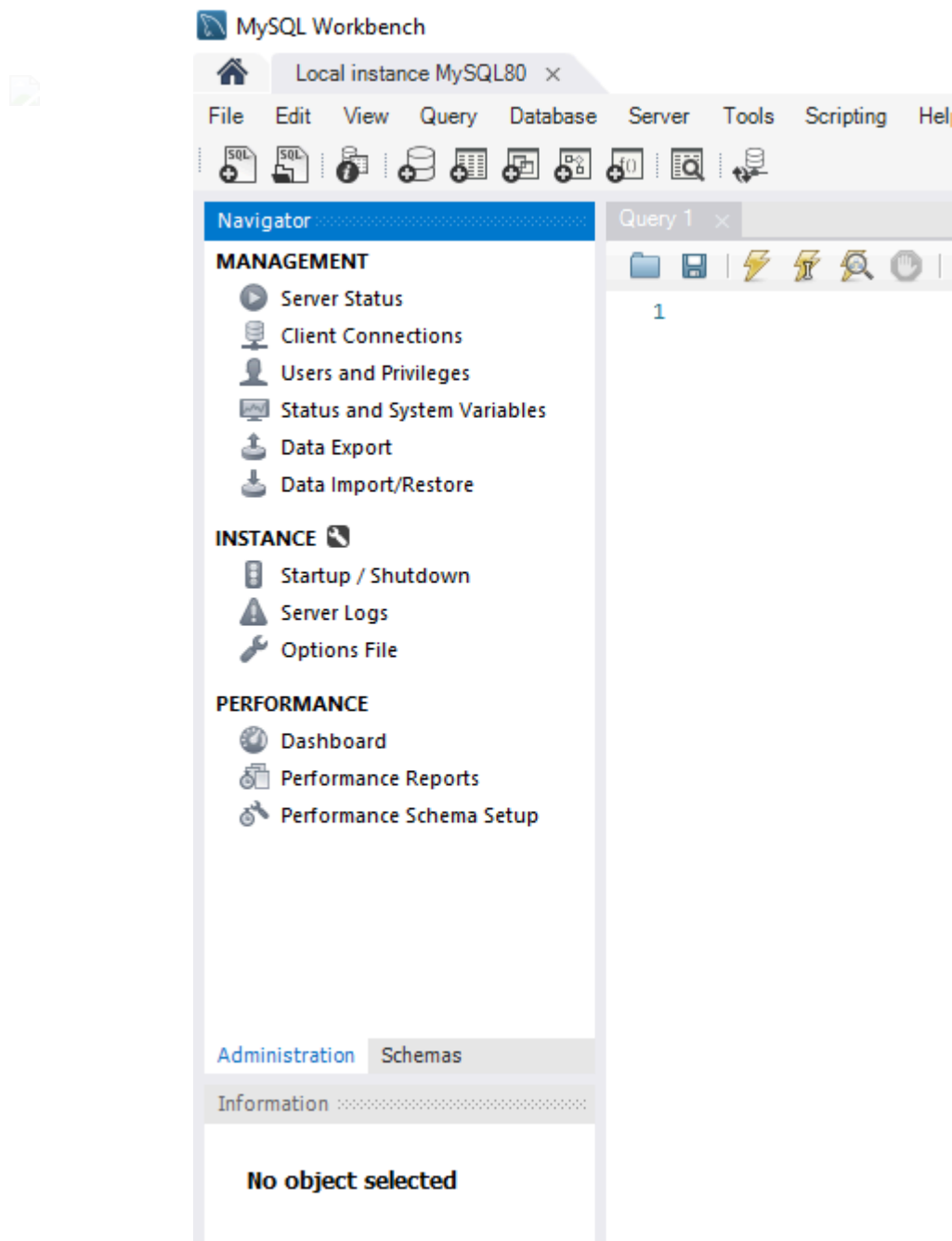


Figura 6 – Aba **Administration** na área **Navigator** do MySQL Workbench

A figura apresenta a tela principal do Workbench, destacando a parte esquerda com a área “Navigator”. Abaixo, nessa área, está acionada a aba “Administration” e desativada a aba “Schemas”. No corpo dessa área há as opções “Server Status”, “Client Connections”, “Status and System Variables”, “Data Export”, “Data Import/Restore”, entre outras.

Na seção **Management** da área **Navigator**, clique em **Data Import/Restore**. Na aba que abrirá, informe no campo **Import from Dump Project Folder** o caminho do *dump* (*backup*) que você quer restaurar, selecione o banco de dados na caixa **Select Database Objects to Import** e depois clique no botão **Start Import**.

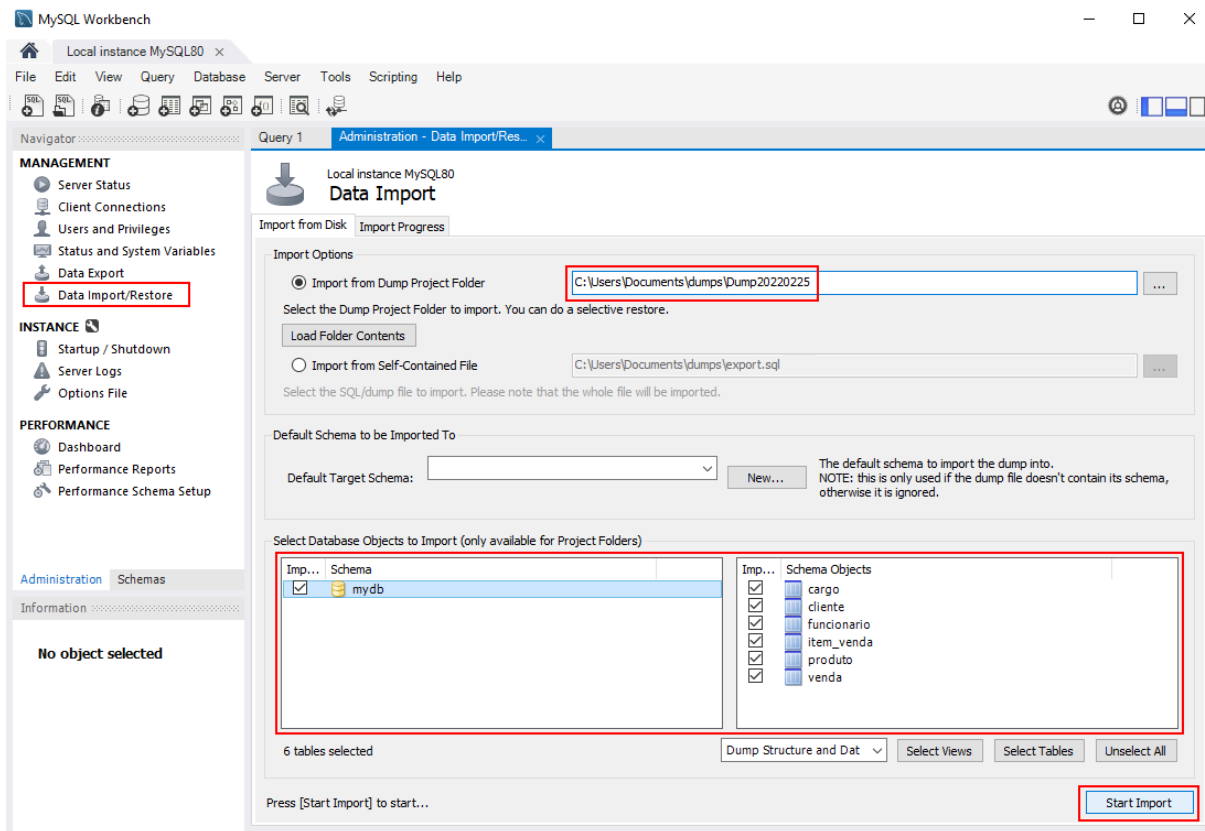


Figura 7 - Restaurando *backup* no Workbench

A figura mostra a janela do programa Workbench 8, e na barra de “Management” lateral à esquerda está circulada em vermelho a opção “Data Import/Restore”. No meio da figura está selecionada a opção “Import from Dump Project Folder” e, na caixa de texto ao lado desta está o caminho da pasta “dump” feita anteriormente durante o backup. Neste caso, o caminho indicado é “C:\Users\vine_\Documents\dumps”. Abaixo, no canto inferior direito, está o botão “Start Import” circulado em vermelho para destaque.

Observe que você tem a opção de importar todo o banco ou apenas tabelas específicas, ou, ainda, se a restauração será de dados e estrutura (**Dump Structure and Data**), apenas de estrutura (**Dump Structure Only**) ou apenas de dados (**Dump**

Data Only). Isso é importante para o caso de você querer importar apenas tabelas que tiveram alteração recente e/ou apenas a estrutura do banco ou suas tabelas.

Atente-se ainda ao fato de que essa operação criará ou sobrescreverá um banco de dados com o nome do banco feito no *backup*; caso você queira apenas uma cópia do banco, pode ser necessário renomear o original. Outra opção é, na exportação, criar um *backup* de arquivo único sem especificação de criação de *schema*.

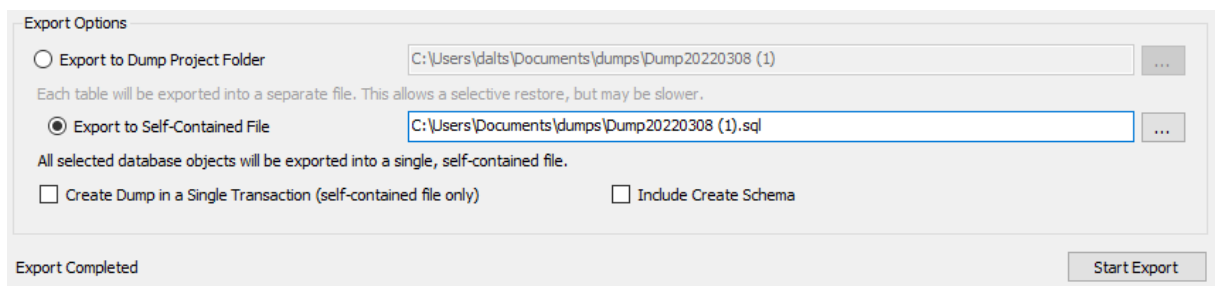


Figura 8 – Criando, na janela de **Export**, um *backup* em arquivo único sem definição de *schema*

A figura mostra detalhes da janela de Export na área “Export Options”, evidenciando a opção “Export to Self-Contained File” e desmarcando a opção “Include Create Schema”.

Ao importar esse *backup*, é possível, então, definir um novo banco de dados para o qual serão enviados a estrutura e os dados. Será preciso definir a opção **Import from Self Contained File** e escolher para qual banco enviar, selecionando o campo **Default Target Schema**. É possível inclusive criar um novo banco pelo botão **New**, que está ao lado desse campo. Veja o exemplo na figura:

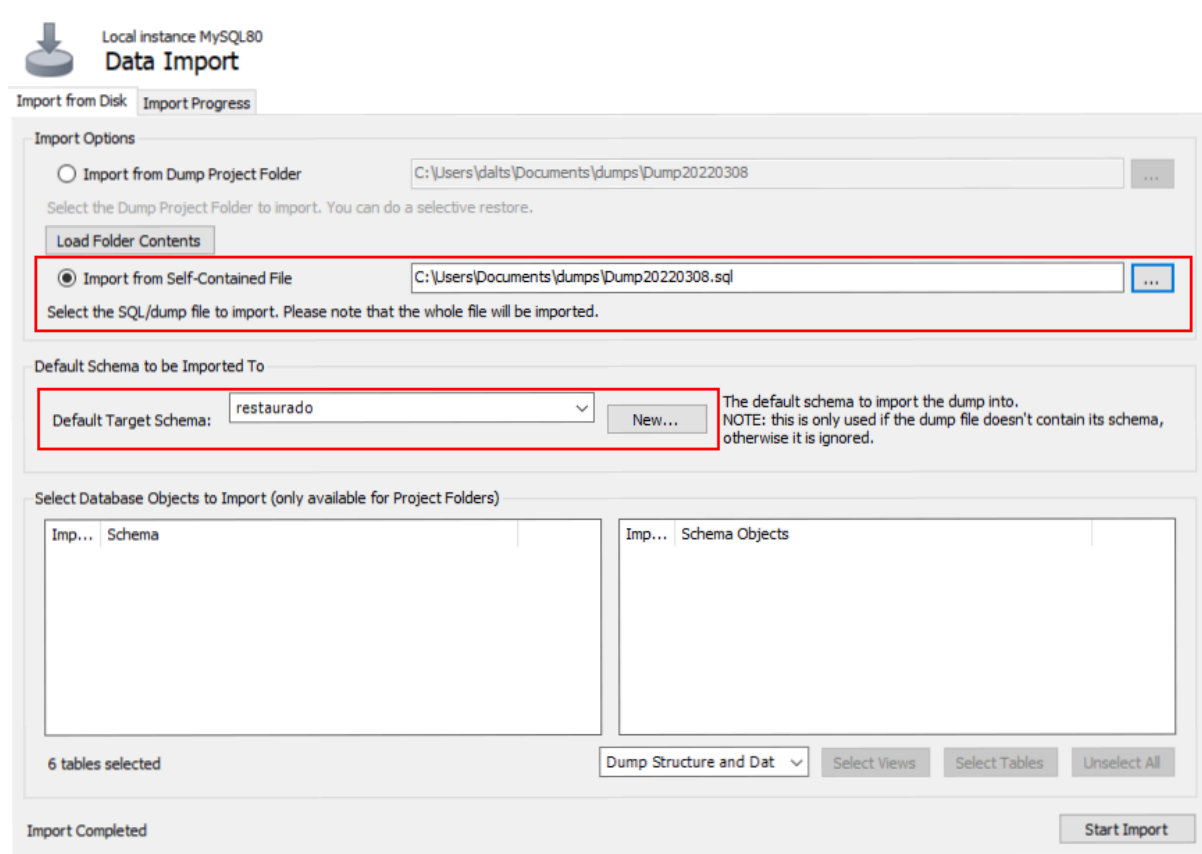


Figura 9 – Importando *backup* de arquivo único para novo banco de dados

A figura mostra detalhes da janela “Import”, destacando a opção “Import from Self-Contained File” selecionada. Abaixo, na área “Default Schema to be Imported To”, destaca-se o campo “Default Target Schema”, no qual está selecionado o valor “restaurado”, e o botão à direita “New...”.

Uma vez restaurado o banco, é possível realizar *refresh* na aba **Schemas** e verificar as alterações nos bancos de dados.

Atente-se sempre para não sobrescrever dados do banco atual, caso tenha dados nele.

SQLBackupAndFTP: ferramenta automatizada

Outra opção viável é utilizar ferramentas de terceiros para *backup* do banco de dados, que tragam funcionalidades e facilidades extras ao processo. Uma dessas ferramentas é o SQLBackupAndFTP, no qual é possível agendar o *backup* total de até dois bancos de dados SQL na versão gratuita do programa.

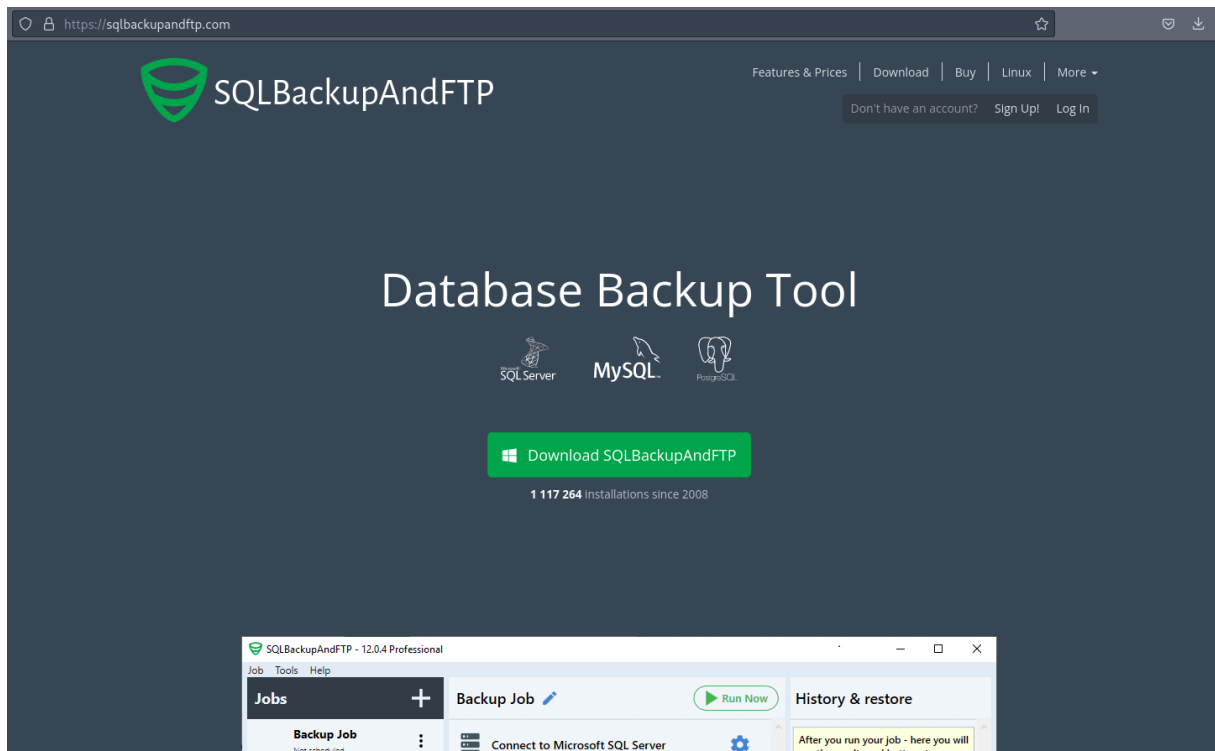


Figura 10 – Site do *software* SQLBackupAndFTP

A imagem mostra a tela inicial do site do aplicativo SQLBackupAndFTP, na qual um botão verde ao centro leva à página de download.

Acessando o *site* do *software* (pesquise por “SQLBackupAndFTP” na *web*), clique no botão de *download* e a versão para Windows será baixada. Para Linux, basta ir no *link* de *downloads* e entrar no *download* da versão Linux, que o levará para a página de *download* do SqlBak, que é a versão para Linux deste aplicativo.

Depois de efetuado o *download*, ao executar o instalador, os próximos passos são simples. Ao abrir o programa, clique no menu em formato de engrenagem à esquerda da faixa **Connect to Database Server** para começar a configurar o seu *backup* automatizado.

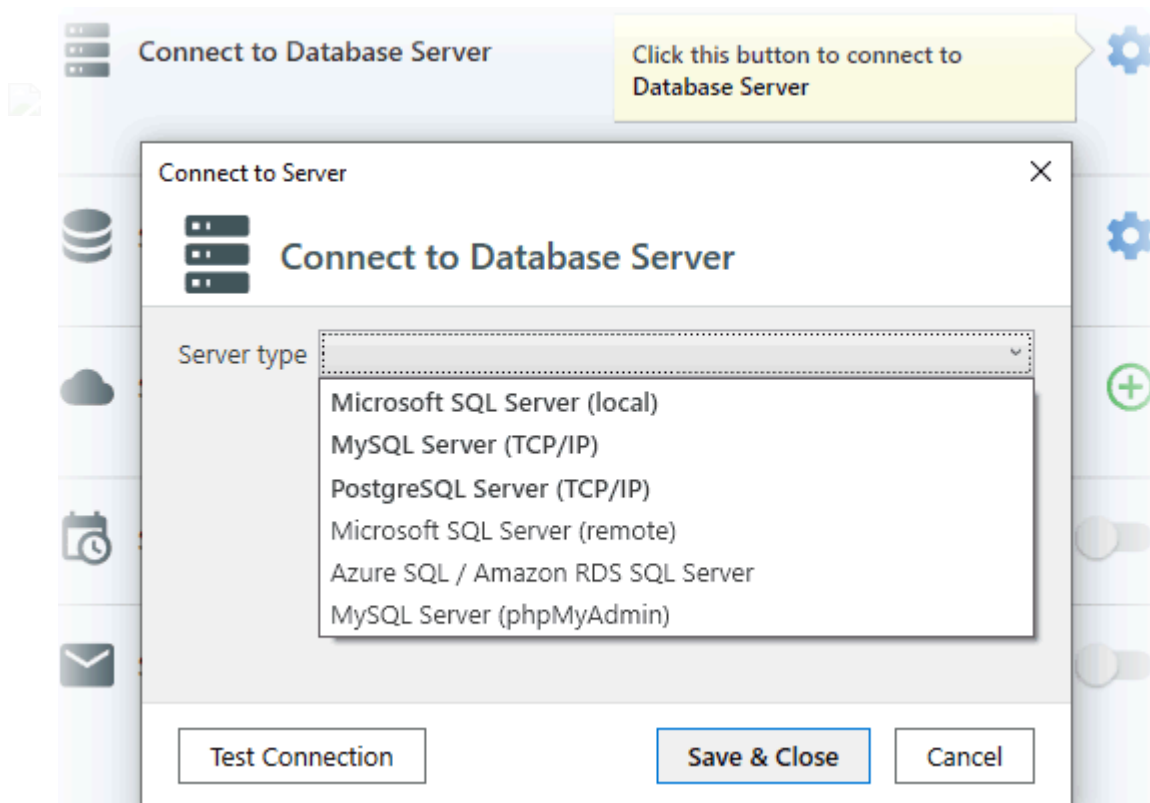


Figura 11 – Escolhendo o tipo de servidor de banco de dados para agendar os *backups*

A figura mostra a janela de seleção de conexão ao servidor de banco de dados. Abaixo estão os botões: à esquerda, “Test Connection”, à direita, “Save and Close” e também “Cancel”.

Em **Server Type**, utiliza-se o MySQL (já que a ferramenta trabalha com vários Sistemas de Gerenciamento de Banco de Dados – SGBDs). Em **Server name**, inclui-se o endereço do servidor – nestes testes, sempre **localhost**. Em seguida, informa-se a senha de **root**. Você pode testar a conexão com o botão que está no canto inferior esquerdo. Salve os dados com o botão **Save and Close**.

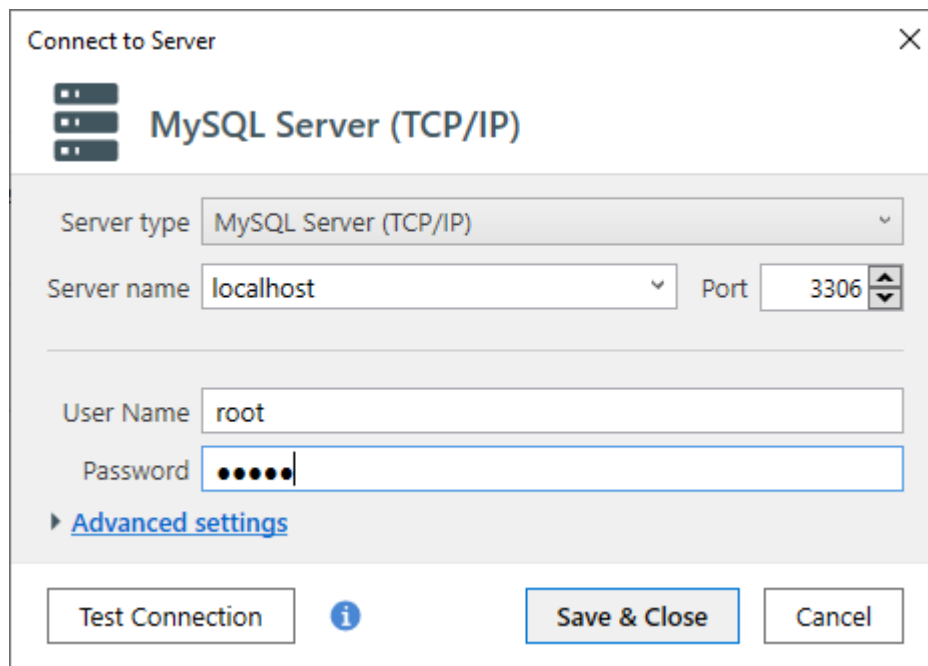


Figura 12 – Configurando a conexão ao servidor de banco de dados

A figura mostra a janela com caixas para preenchimento de “Server name”, “Port” (valor padrão 3306), “User Name” (valor padrão “root”) e Password. Abaixo, à esquerda há o botão “Test Connection” e, à direita, os botões “Save and Close” e “Cancel”.

O próximo passo é acionar o ícone de engrenagem da faixa **Select Databases** para poder escolher de quais bancos se deseja fazer o *backup*.

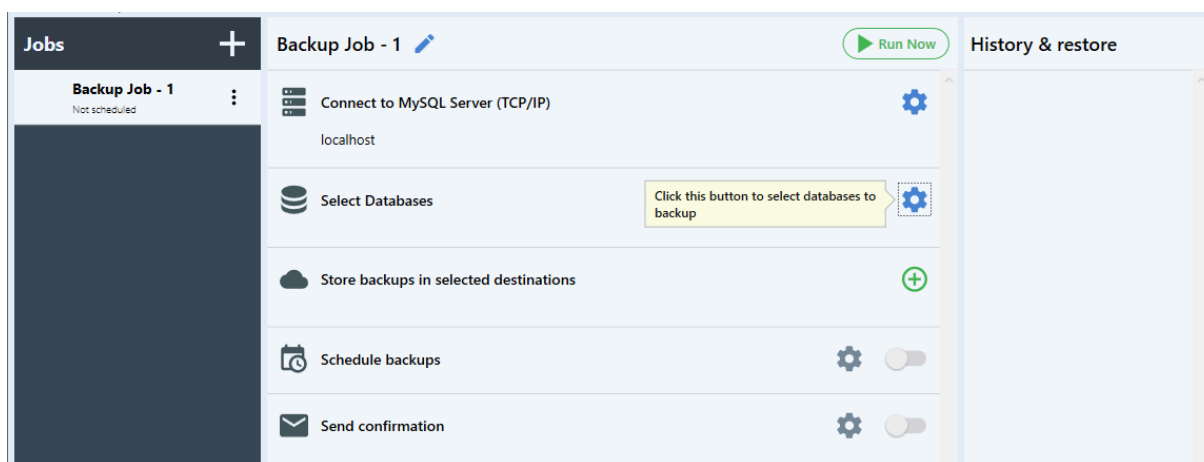


Figura 13 – Passo de seleção do banco de dados que terá *backups* realizados automaticamente

A figura mostra a tela principal do programa, destacando a faixa “Select Databases” e seu ícone de engrenagem à direita.

Na janela que se abre, é possível marcar cada um dos bancos de dados desejados. Caso você marque a opção **Backup all non-system databases**, todos os bancos criados posteriormente a esse *backup* serão incluídos na próxima vez que o *backup* for efetuado. Isso proporciona mais segurança, caso novos bancos sejam criados.

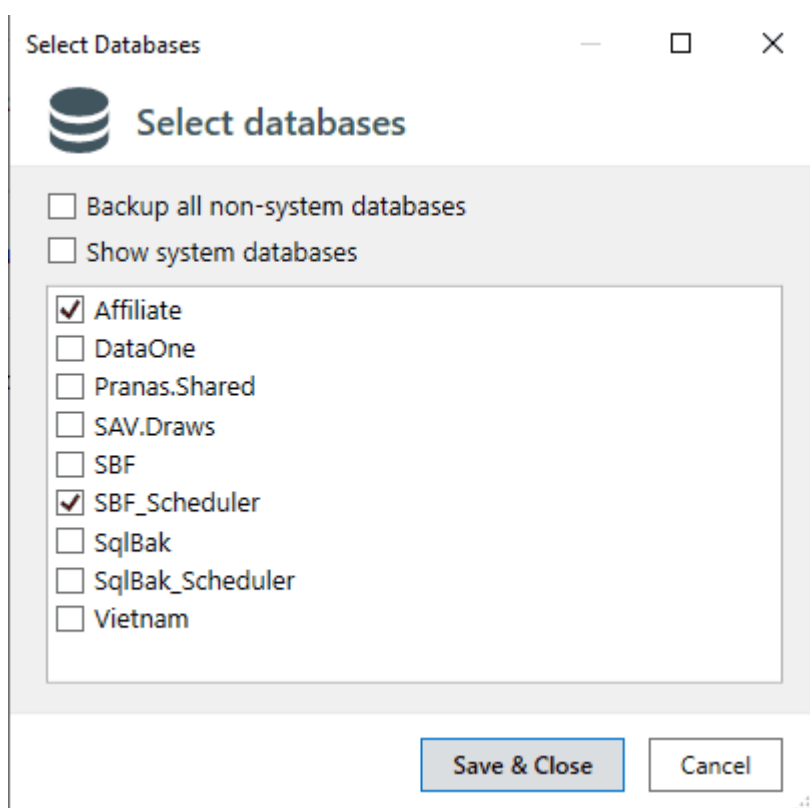


Figura 14 – Selecionando os bancos de dados para *backup*

A imagem mostra a janela para seleção dos bancos dos quais será feito o backup. Ao centro estão as caixas de seleção com a lista de todos os bancos disponíveis para o backup, entre os quais dois estão selecionados. Acima estão mais duas caixas de seleção: “Backup all non-system databases” e “Show system databases”.

No próximo passo da criação do *backup*, você pode escolher onde serão salvos os arquivos de *backup*.

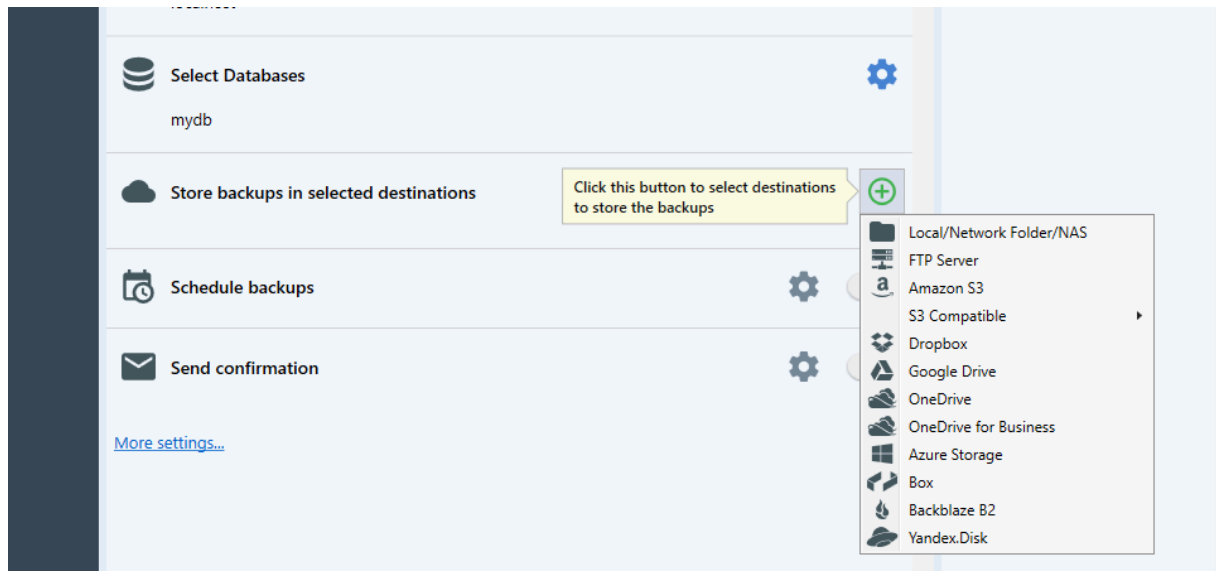


Figura 15 – Selecionando onde o *backup* será efetuado

A figura mostra, acima e à direita, um sinal de “+” em verde, para o qual há um menu dropbox listando serviços de nuvem e armazenamento local, como Local/Network Folder/NAS, FTP Server, Amazon S3 Compatible, Dropbox, Google Drive etc.

Note que, ao clicar no ícone “+”, aparecem várias opções de locais para salvar o *backup*. Como, neste caso, este é um *backup* local, pode-se optar por salvar na rede interna ou na máquina própria, ou utilizar serviços na nuvem, entre os quais vários contêm planos gratuitos, como o OneDrive e o Google Drive.

Caso queira agendar o *backup* para que ele seja efetuado rotineiramente, é possível escolher o horário e a frequência. Para isso, você deve ativar o botão na faixa **Schedule backups** e informar o horário.

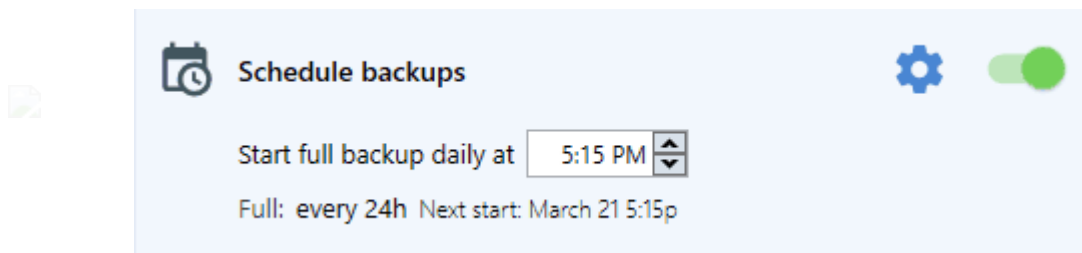


Figura 16 – Janela de seleção de horário e agendamento de *backups* diários

A imagem mostra uma pequena janela de nome “Schedule backups”, que está acima e à esquerda. Ao centro da janela está uma caixa de seleção de horário com o horário 5:15 PM e com a descrição dessa seleção: “Start full backup daily at”. Abaixo disso está o texto: “Full: every 24h Next start: March 21 5:15p”.

Por padrão, o agendamento será diário, mas, a partir do ícone de engrenagem dessa faixa, será possível ajustar a periodicidade.

Outra função desse *software* que pode ser útil é a capacidade de enviar um *e-mail* de confirmação para cada *backup* criado, informando sucesso ou falha deste. Essa função pode ser acionada na faixa **Send confirmation** e configurada com mais detalhes no ícone de engrenagem dessa faixa.

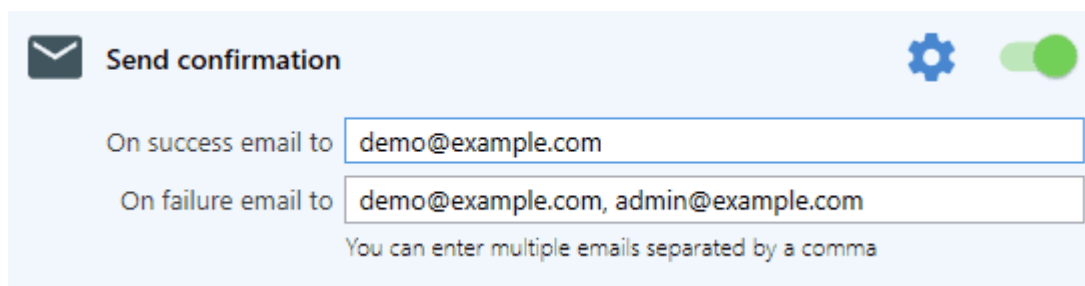


Figura 17 – Configurando o envio de *e-mail* de confirmação de *backup*

A figura mostra a janela “Send confirmation”, na qual há dois campos de texto de nomes: “On success email to” e “On failure email to”. Nesses campos estão dois endereços de e-mail, sendo o primeiro <demo@example.com>. Acima e à direita constam dois símbolos, um de engrenagem e um de seletor de ligado/desligado na posição ligado.

Por fim, será possível realizar um *backup* imediatamente, clicando em **Run Now** no topo da janela. O *backup* será armazenado na pasta informada nas configurações de **Store backups**.

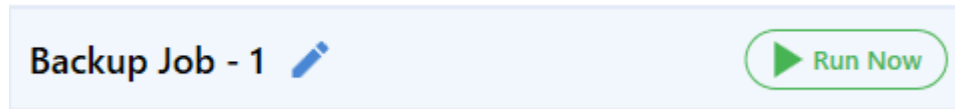


Figura 18 – Janela com o botão **Run Now** para efetivar e começar o *backup*

A imagem mostra uma pequena janela em que, à esquerda, está o texto “Backup Job - 1”, com o símbolo de um lápis, que serve para a edição do nome deste job” e, à direita, está o botão verde com o texto “Run Now”.

Restauração com SQLBackupAndFTP

Com a ferramenta SQLBackupAndFTP, todos os seus *backups* podem ser encontrados na seção histórico e restauração (**History and restore**). Você pode abrir um *log* de *backup* clicando em **Open Log**, baixar arquivo(s) de *backup* clicando em **Download Backup** ou restaurar o banco de dados do *backup* clicando em **Restore from Backup**.

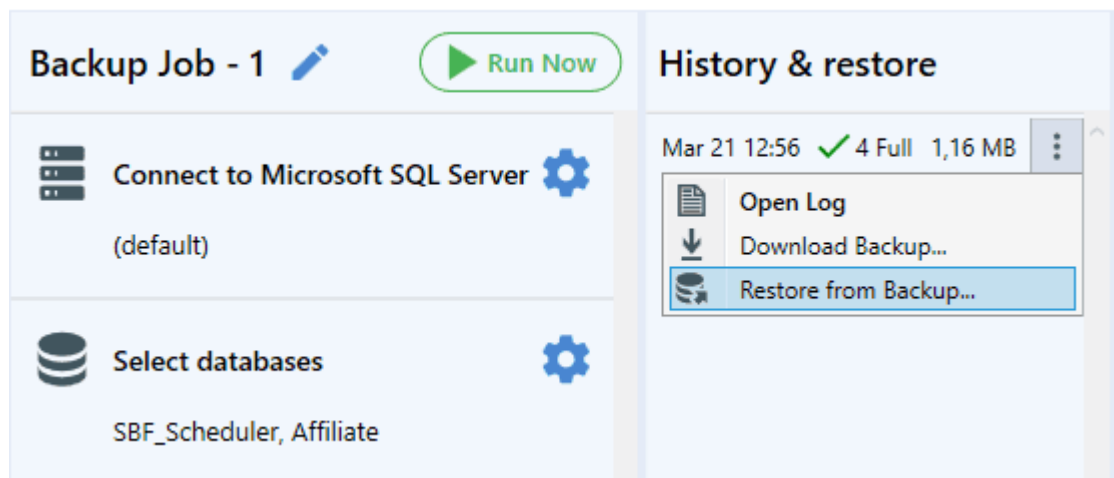


Figura 19 – Restaurando um *backup* anterior

A figura mostra a janela de restauração de um backup anterior (“History and restore”). Nela, à direita, no menu sandwich (menu dos três pontinhos) que está clicado, está o menu drop down com as opções “Open Log”, “Download Backup” e “Restore from backup”. Esta última está selecionada.

A opção **Open Log** mostra o histórico de *backup* de seus computadores (essa opção está disponível apenas na versão paga do aplicativo).

Jobs ↻

order by: Job name ↓ | Computer | Status | Last run

✓ Backup Job - 1
DESKTOP-DELL ⌚ an hour ago

History ↻

Job Backup Job - 1 Computer DESKTOP-DELL

When	Status	Size
Mar 21 12:56 PM	✓ 4 Full	2.33 MB

*WebLog data is available for the last 30 days for up to 1000 runs for each job

Figura 20 – Confirmação de *backup* efetuado com sucesso e lista de *backups*

A figura mostra a janela de “Jobs” (trabalhos de backup), em que está, acima e ao centro, o texto “Backup Job - 1 DESKTOP-DELL”, com o “V” em verde, confirmando o backup efetuado. Abaixo disso está a janela “History”, listando todos os backups efetuados. No caso, a lista tem apenas um backup com a data e hora mostradas na imagem 19.

Para restaurar um *backup* anterior, clique em **Restore from Backup** e escolha o arquivo anteriormente criado pelo serviço de *backup*.

Automatização via sistema operacional



Nem sempre você terá um *software* completo que automatize rotinas de *backup*. Algumas limitações podem impor-se e, nesse caso, vale recorrer a automatizações próprias e nativas dos sistemas operacionais. A seguir, confira exemplos para Linux e Windows.

É possível que haja a necessidade de configurar o caminho de instalação do MySQL nas variáveis de ambiente do sistema operacional, para que a linha de comando seja executada corretamente.



Automatizando uma rotina de *backup* em servidores Linux

De maneira muito simples, você pode criar um *bash script* para fazer o *backup* de um banco de dados. Abra seu editor de texto favorito (por ex.: Gedit, Vim, Mousepad etc.) e digite o texto abaixo (após as barras “//”, são apenas comentários e não precisam ser digitados).



#!/bin/sh	
FILE=backup.sql.`date +"%Y%m%d"``	// Aqui se cria a variável FILE , que receberá o nome do arquivo de <i>backup</i> com a data do dia em que foi criado.
DBSERVER=127.0.0.1	// Aqui se insere o endereço IP do servidor. No caso, o <i>server</i> é local e está no IP local.
# DBSERVER=localhost	// Caso esteja configurado nos <i>hosts</i> da máquina, em vez do IP, pode-se utilizar o localhost .
DATABASE=database-name	// Nessa linha, insere-se o nome da base de dados.
USER=user-name	// Aqui, insere-se o nome de usuário da base de dados: pode ser root .
PASS=your-password	// Insere-se a senha do usuário inserido na linha anterior.
unalias rm 2> /dev/null rm \${FILE} 2> /dev/null rm \${FILE}.gz 2> /dev/null	// Nessa linha e nas próximas duas, garante-se que não tenha um arquivo já com o nome criado na pasta em que o <i>backup</i> será feito.
mysqldump --opt -- user=\${USER} -- password=\${PASS} \${DATABASE} > \${FILE}	// Executa-se então o programa mysqldump , que será responsável por executar o seu <i>backup</i> . Note que ele já usa as variáveis USER e PASS , criadas anteriormente.
gzip \$FILE	// Comprime-se o arquivo com o Gzip.
echo "\${FILE}.gz was created:"	//Avisa-se o usuário que o arquivo foi criado.
ls -l \${FILE}.gz	// Lista-se então o arquivo criado.

Salve o arquivo com o nome de “local-backup.sh”.

Após a criação do *script*, é preciso dar permissão de execução para que se possa executá-lo como um programa.

Crie uma pasta para colocar o *script* e os *backups*.

<code>mkdir -p /backup/mysql</code>	// Aqui se cria a pasta.
<code>cd /backup/mysql</code> # nesta pasta, colocamos o nosso script criado acima	// Abra a pasta.
<code>chmod +x /backup/mysql/local-backup.sh</code>	// Nessa linha, é preciso dar permissão de execução para o seu <i>script</i> .

Com esse simples *script*, você pode criar o seu arquivo *dump* de *backup*.

Como automatizar o *backup* para que ele seja executado todo dia?

Em sistemas Linux, essa tarefa é bem simples e pode ser feita utilizando o **Crontab**, uma função do sistema Linux que serve para agendar tarefas diárias.

Acesse o arquivo do Crontab com o comando:

```
crontab -e
```

Dentro desse arquivo, insira a seguinte linha:

```
0 1 * * * /backup/mysql/local-backup.sh 1>> /var/log/mysqlbackup.log  
2>>/var/log/mysqlbackup-error.log
```

Note que **/backup/mysql/local-backup.sh** é o caminho do *script* de *backup*.

Com essa linha adicionada ao Crontab, o *backup* será efetuado todos os dias à meia-noite. O primeiro 0 da linha indica que o *backup* será executado à meia-noite, depois será informado qual *script* será executado: no caso, é o *script* criado

anteriormente. Os próximos dois comandos são para que, em caso de algum erro, o *backup* seja salvo no arquivo de *log* de erros.



Automatizando uma rotina de *backup* em servidores Windows

No Windows, pode-se utilizar um *script* em BAT. Abra o então o bloco de notas do Windows e insira o seguinte:

<code>echo off</code>	// Limpa-se a linha para a execução do <i>script</i> .
<code>set currrdate=%date:~4%</code>	// Cria-se uma variável com a data atual para ser utilizada para salvar o arquivo.
<code>set filedate=%currrdate:/_=%</code>	// Cria-se uma variável com a data atual para ser utilizada para salvar o arquivo.
<code>"C:\Program Files\MySQL\MySQL Workbench 8.0 CE\mysqldump.exe" -u [Seu usuario do banco de dados] -p[Sua senha] [O nome do banco de dados] > "[O caminho onde deseja salvar o backup ex: D:\backups]_%filedate%.sql"</code>	

Substitua os valores de caminho de arquivo, usuário e senha pelos dados correspondentes à sua realidade. Salve o arquivo com o tipo “todos os arquivos” e com o nome “backupscrip.bat”.

Para automatizar a execução diária do *script*, utilize o Task Scheduler do Windows (principalmente no caso do Windows 10).

A busca em português nem sempre retorna o programa correto, sendo mais efetivo buscar o termo em inglês *task scheduler*.

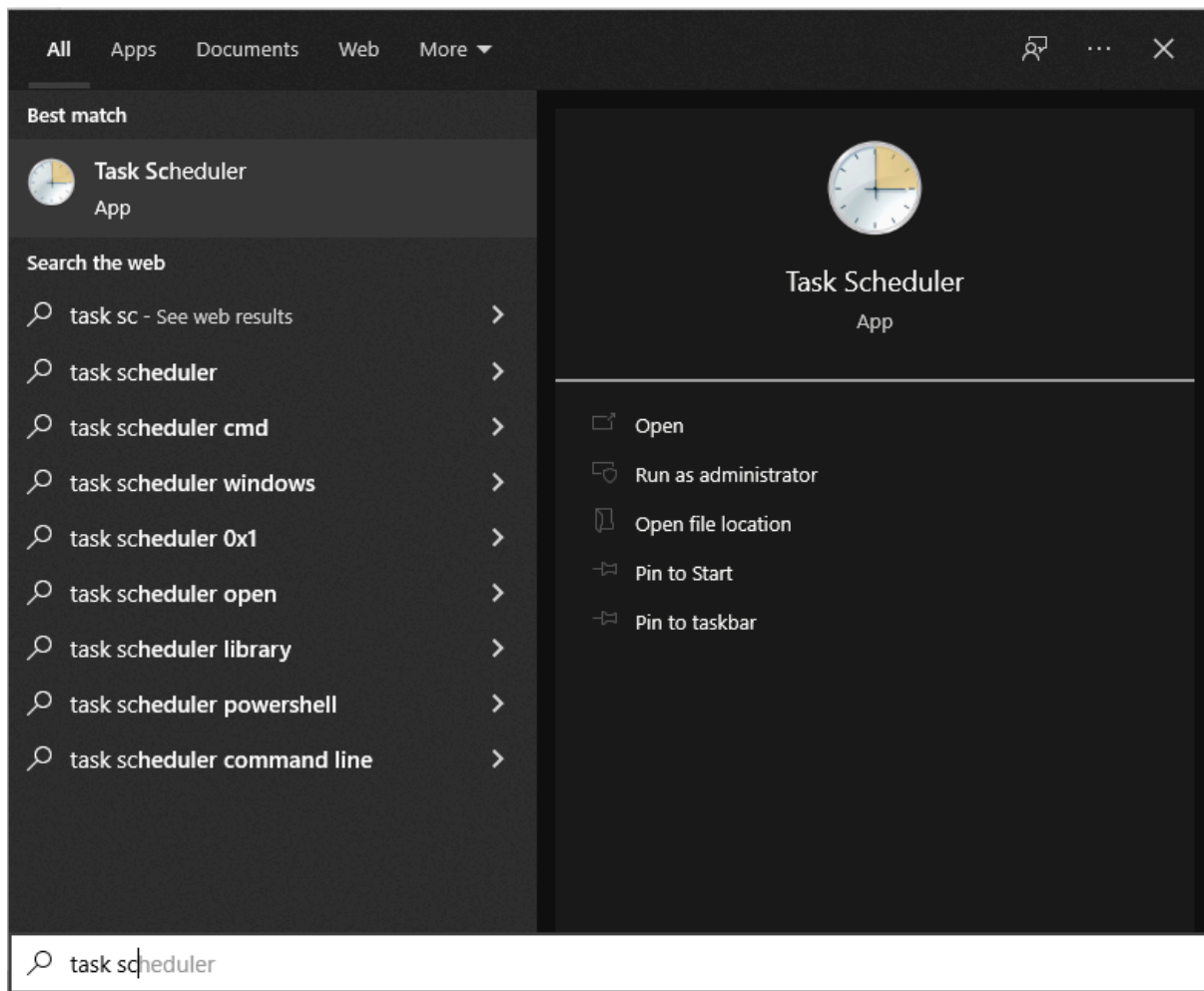


Figura 21 – Busca em inglês do programa Task Scheduler

Já no Task Scheduler, clique com o botão direito do *mouse* em **Task Scheduler Library** e escolha uma nova pasta (**New Folder**).

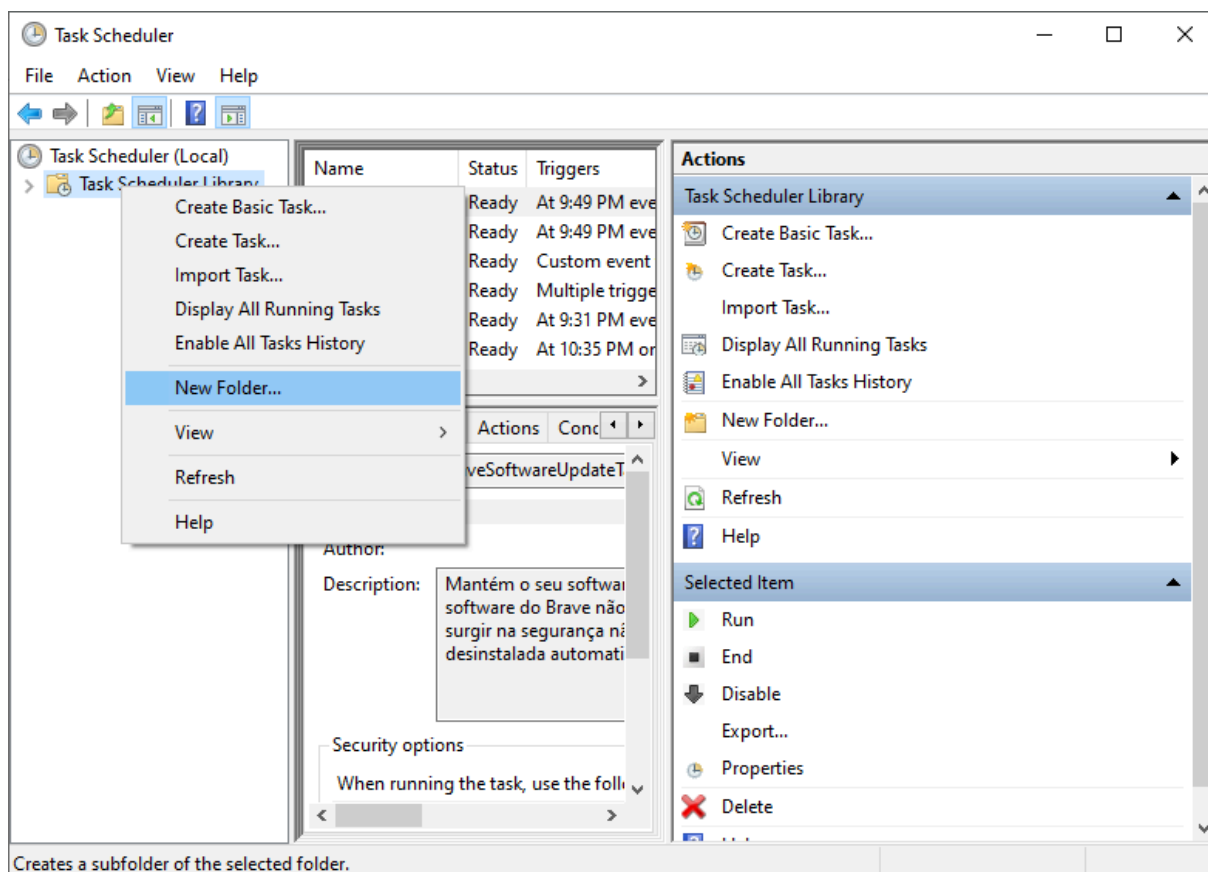


Figura 22 – Escolhendo uma nova pasta (**New Folder**)

Em seguida, insira o nome na pasta. Nesse caso, será o utilizado o nome **Meus Scripts**.

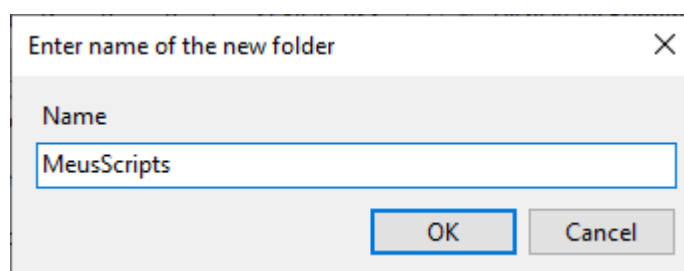


Figura 23 – Inserindo o nome na pasta

Clique então com o botão direito na pasta criada e selecione a opção **Create a Basic Task**, que levará você para a seguinte janela:

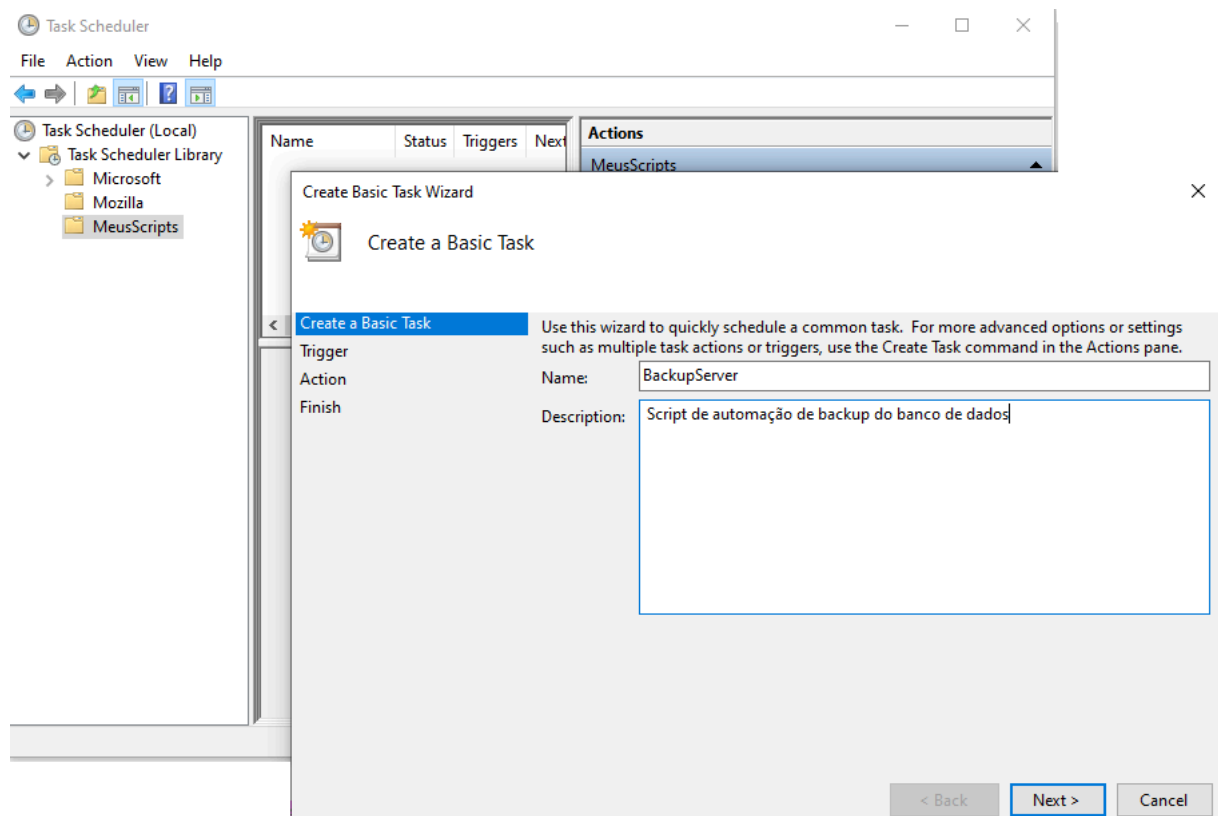


Figura 24 – Janela de criação de tarefas

Depois de clicar no botão **Next**, você será levado à próxima janela:

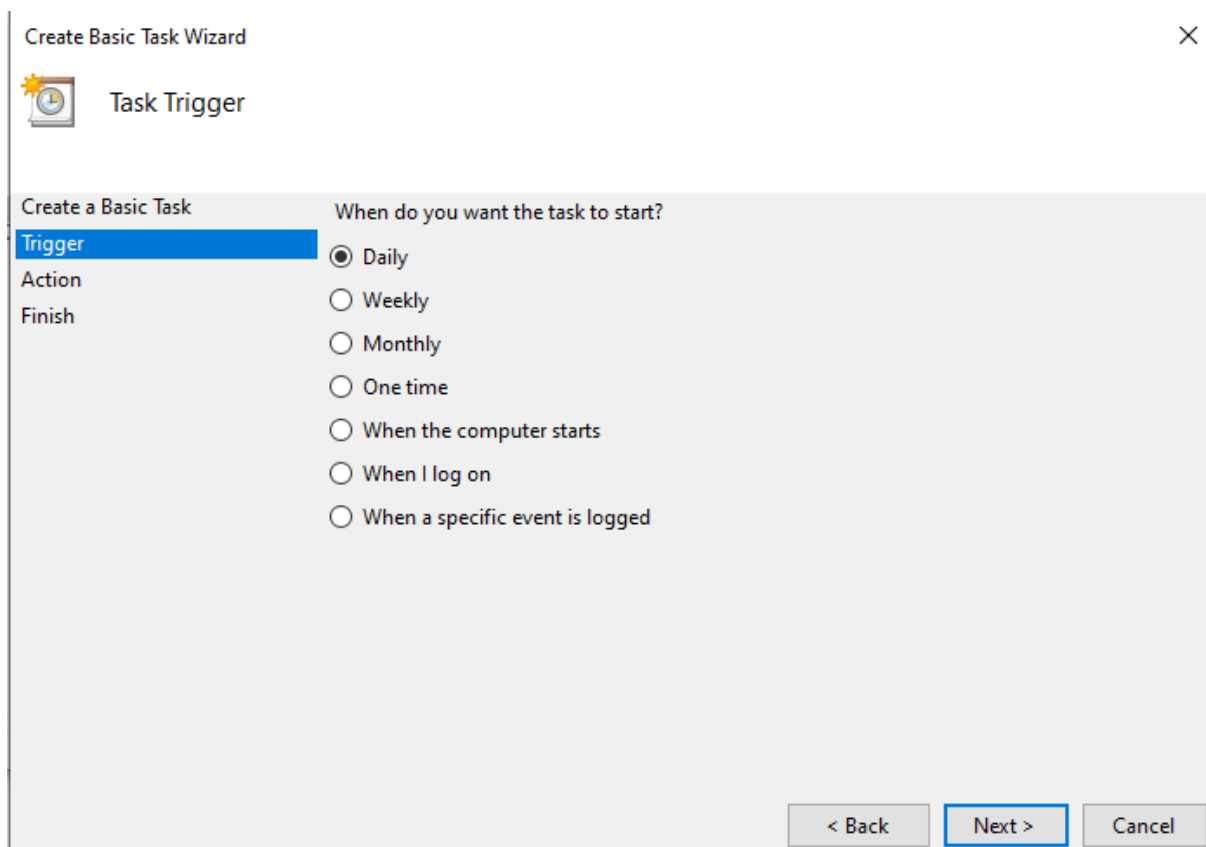


Figura 25 – Escolhendo a opção diariamente (**Daily**)

Depois de selecionar a opção **Daily** (diariamente), clique em **Next**.

Na próxima janela, escolha quando você quer que comece a execução da tarefa e por quantos dias ela deverá ser executada. Nesse caso, a escolha foi por um ano (365 dias).

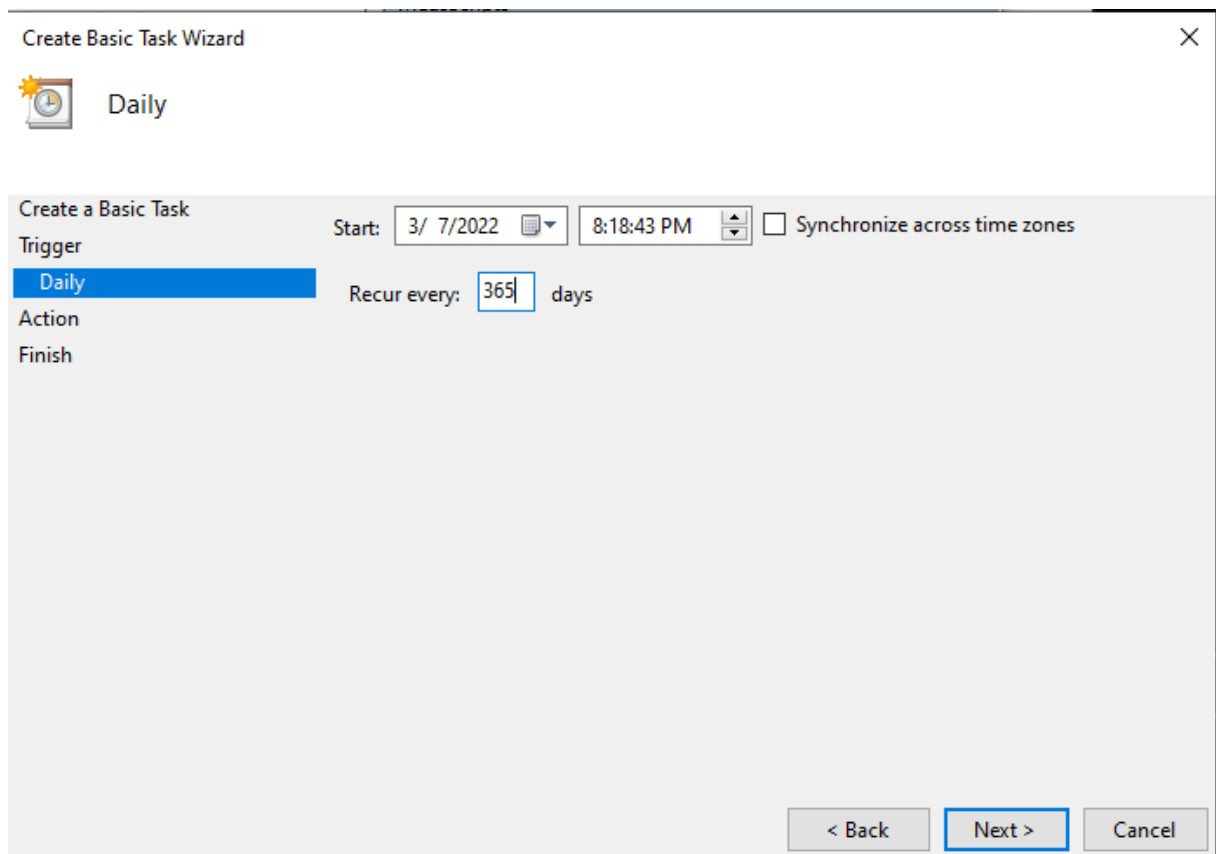


Figura 26 – Escolhendo a data inicial da tarefa e por quanto tempo ela será executada

Na próxima janela, escolha a opção de iniciar um programa (**Start Program**).

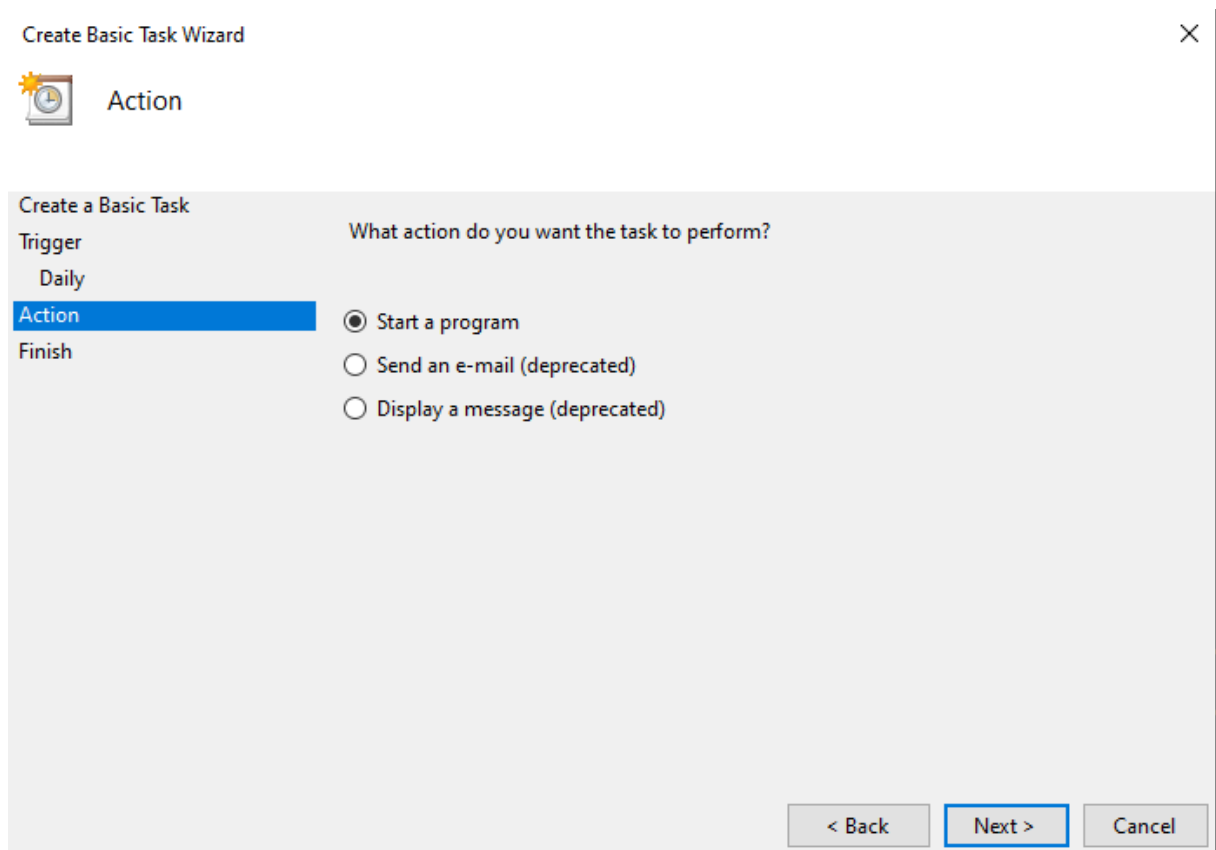


Figura 27 – Selecionando a opção de iniciar um programa

Após isso, selecione então o caminho do *script* que deverá ser executado.

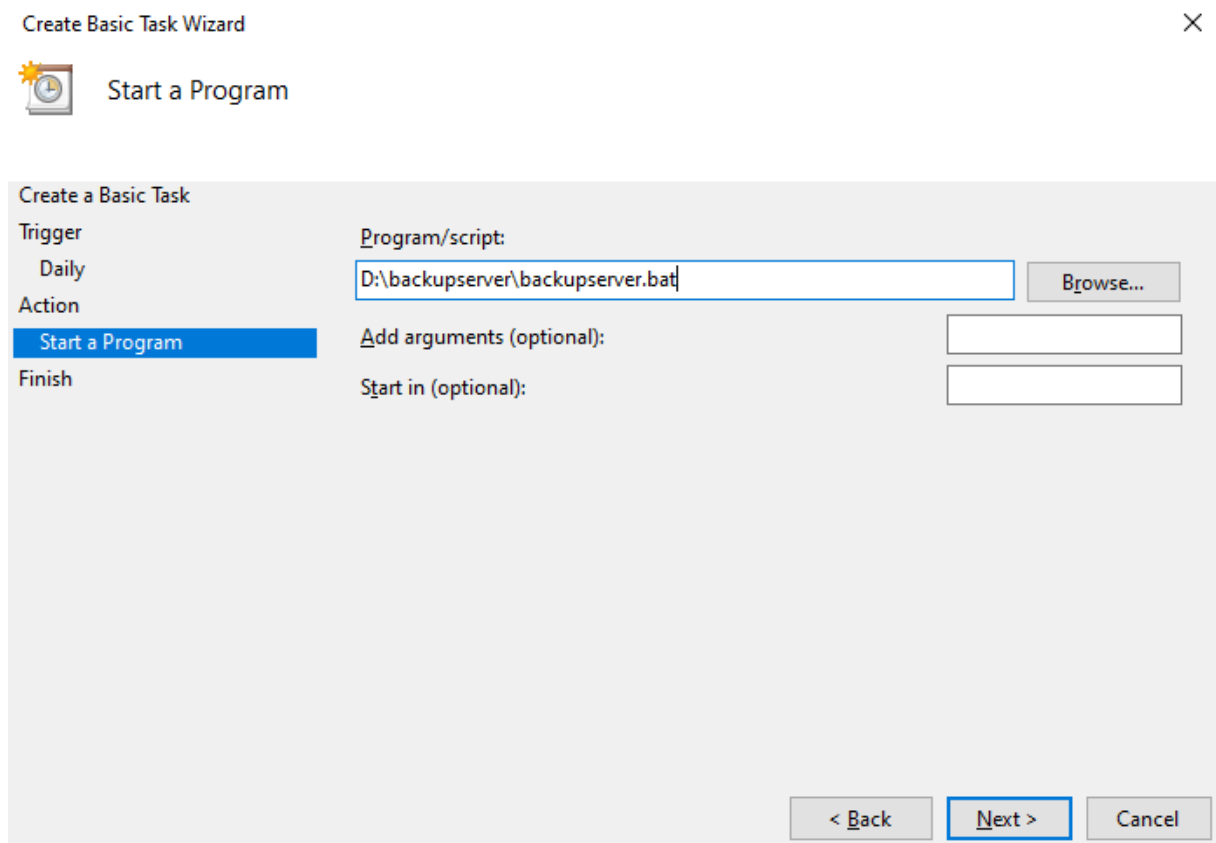


Figura 28 – Selecionando o caminho do *script*

Por fim, confirme as informações da sua tarefa diária de *backup* e clique em **Finish**.

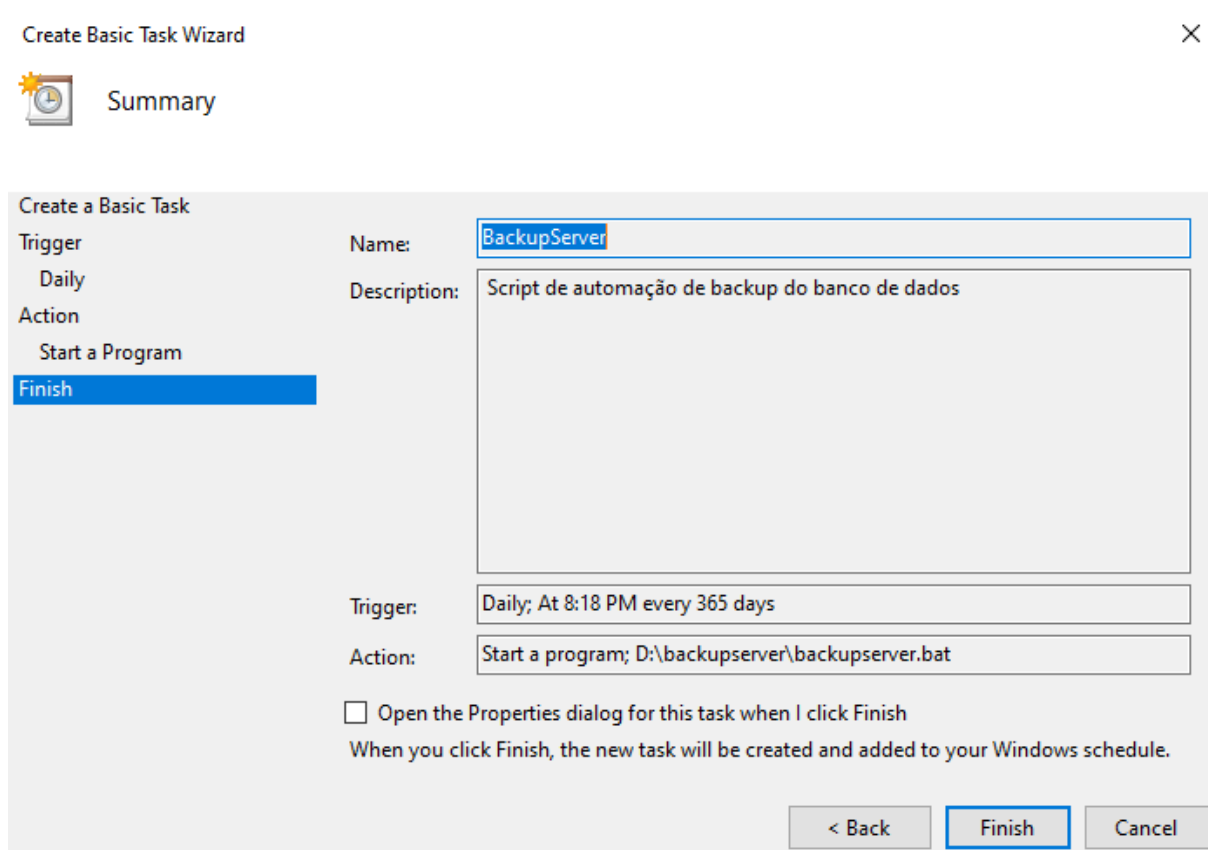


Figura 29 – Confirmação, finalização e ativação da tarefa

Encerramento

Neste conteúdo, você estudou que planos de desastres, *backups* e restauração de dados são as mais importantes tarefas de um administrador de bancos de dados. Com ferramentas como o MySQL Workbench, o SQLBackupAndFTP e um pouco de conhecimento em linguagens de *script*, você poderá automatizar e manter seguros os seus dados.

Para cada cenário de perda de dados há uma reação e um impacto diferente, por isso é sempre importante manter os seus planos de recuperação de dados abrangentes e cobrindo o máximo de cenários “genéricos” possível.

Nunca se sabe quando um desastre natural pode ocorrer ou quando aquela xícara de café pode cair sobre o equipamento. Portanto, seus problemas serão sempre menores, caso você esteja preparado para tudo.

