



ENTER

Curso Técnico em Desenvolvimento de Sistemas Online

**BANCODE DADOS II**

# **GEEaD - Grupo de Estudo de Educação a Distância**

## **Centro de Educação Tecnológica Paula Souza**

### **Expediente**

*GEEaD – CETEC  
GOVERNO DO ESTADO DE SÃO PAULO  
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO  
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS  
FUNDAMENTOS DE INFORMÁTICA*

*Autor: José Mendes Neto*

*Revisão Técnica: Eliana Cristina Nogueira Barion  
Lilian Aparecida Bertini*

*Revisão Gramatical: Juçara Maria Montenegro Simonsen Santos*

*Editoração e Diagramação: Flávio Biazim*

# APRESENTAÇÃO

**Este material didático do Curso Técnico em Desenvolvimento de Sistemas modalidade EaD foi elaborado especialmente por professores do Centro Paula Souza para as Escolas Técnicas Estaduais – ETECs.**

**O material foi elaborado para servir de apoio aos estudos dos discentes para que estes atinjam as competências e as habilidades profissionais necessárias para a sua plena formação como Técnicos em Desenvolvimento de Sistemas.**

**Esperamos que este livro possa contribuir para uma melhor formação e aperfeiçoamento dos futuros Técnicos.**

# AGENDA 3

**COMANDOS SQL  
E LINGUAGEM  
DE DEFINIÇÃO  
DE DADOS – DDL**







MERGULHANDO  
NO TEMA...

Como já mencionado na Agenda 1, todos os SGBDs possuem sua interface de comando, com todos os recursos disponíveis do SGBD que irá utilizar.

Mas sem dúvida, a utilização de uma interface gráfica ajuda muito na produtividade de qualquer projeto, assim como utilizamos a ferramenta brModelo para desenvolvimento dos projetos Conceitual e Lógico, utilizaremos o Workbench para implementação de projetos físicos.

Apesar da ferramenta disponibilizar uma série de recursos, vamos utilizar nessa e nas demais agendas somente aqueles relacionados à parte da Linguagem SQL, que possui comandos de DDL (Data Definition Language – Linguagem de Definição de Dados), DML (Data Manipulation Language – Linguagem de Manipulação de Dados) e DQL (Data Query Language – Linguagem de Consulta de Dados).

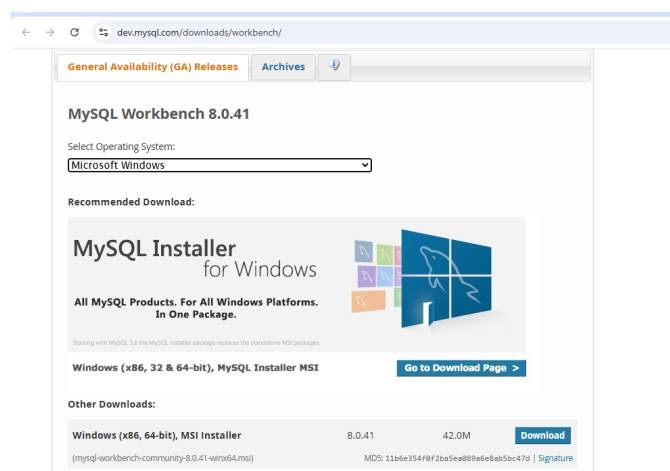
A parte DDL da Linguagem SQL, você já conheceu na agenda anterior. Nessa agenda, vamos aproveitar para revisar esse conteúdo, mas agora utilizando uma interface gráfica.

Vamos começar!!!

## Instalação da Interface Gráfica Workbench

Agora que já instalamos o XAMPP e aprendemos a utilizar o phpMyAdmin para manipulação do banco de dados, vamos instalar o MySQL Workbench, uma ferramenta gráfica oficial do MySQL que facilita a modelagem e a execução de comandos SQL.

Acesse o site <https://dev.mysql.com/downloads/workbench/> para download.



Selecione a versão correspondente a versão de Sistema Operacional utilizado para download. Clique em Download e, se solicitado, selecione **"No thanks, just start my download"** para baixar sem necessidade de login.

## MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »

using my Oracle Web account

Sign Up »

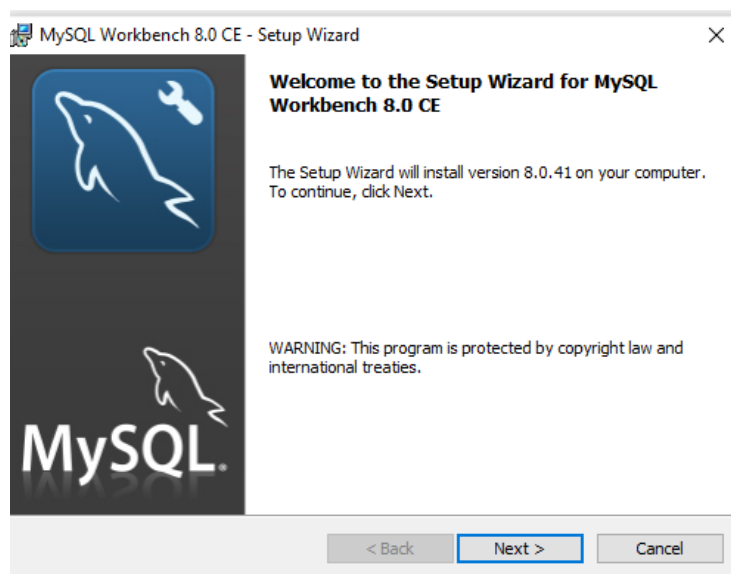
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

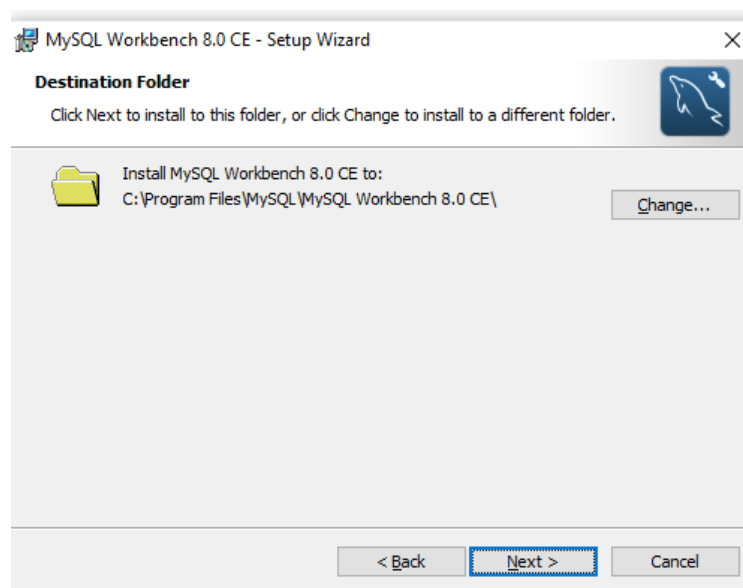
[No thanks, just start my download.](#)

Após download, execute o arquivo.

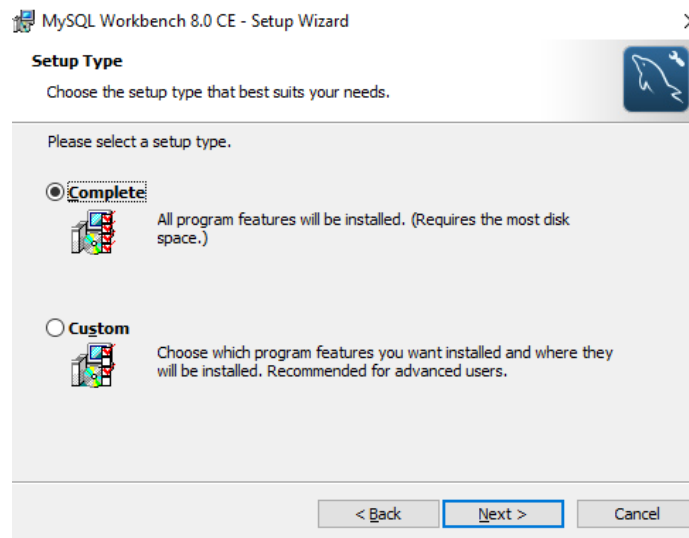
Na tela de boas-vindas, clique em **Next**.



Selecione o local de armazenamento e clique em **Next**.



Escolha a opção "**Complete**" para instalar todos os recursos necessários e clique em **Next**.

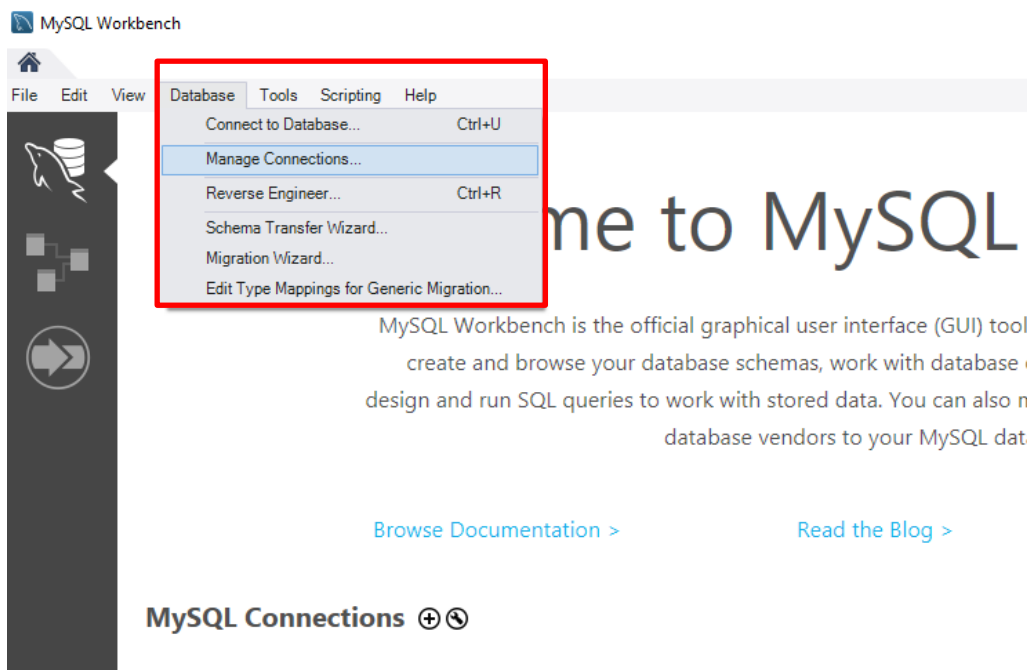


Clique em **Install** e aguarde a instalação.

Após finalizar instalação, como já temos o MySQL instalado via XAMPP, devemos configurar o Workbench para se conectar ao banco de dados existente, evitando conflitos.

Abra o MySQL Workbench.

No menu **Database**, clique em **Manage Connections**.



Clique no botão **New** e preencha os campos:

Manage Server Connections

MySQL Connections  
new connection

Connection Name:

Connection Method:  Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

Password:  Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema:  The schema to use as default schema. Leave blank to select it later.

New Delete Duplicate Move Up Move Down Test Connection Close

**Connection Name:** MySQL\_XAMPP

**Hostname:** localhost

**Port:** 3306

**Username:** root

**Password:** Deixe em branco (a menos que tenha definido uma senha)

Manage Server Connections

MySQL Connections  
new connection  
MySQL\_XAMPP

Connection Name:

Connection Method:  Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

Password:  Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema:  The schema to use as default schema. Leave blank to select it later.

New Delete Duplicate Move Up Move Down Test Connection Close

Clique em **Test Connection** para verificar que está funcionando corretamente.



MySQL Workbench



Successfully made the MySQL connection

Information related to this connection:

Host: localhost  
Port: 3306  
User: root  
SSL: not enabled

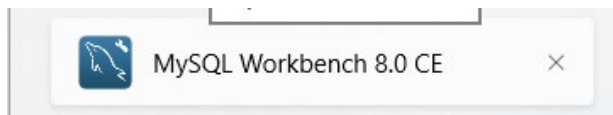
A successful MySQL connection was made with the parameters defined for this connection.

OK

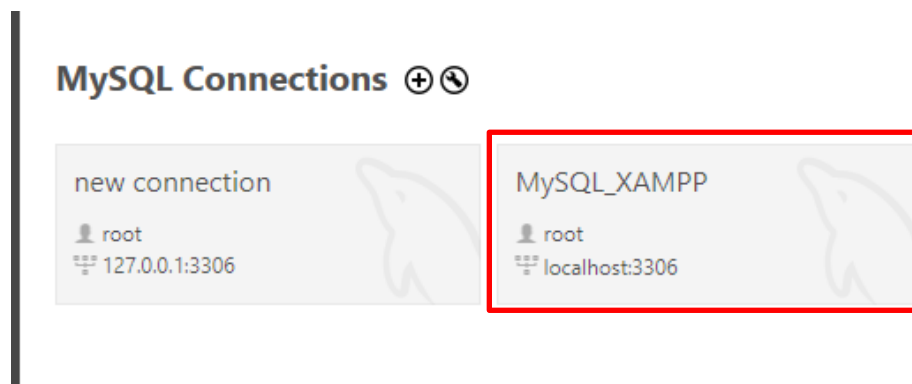
Se o teste for bem-sucedido, clique em **OK** e **Close**.

**IMPORTANTE:** Para testar o funcionamento o XAMPP deve estar ativo.

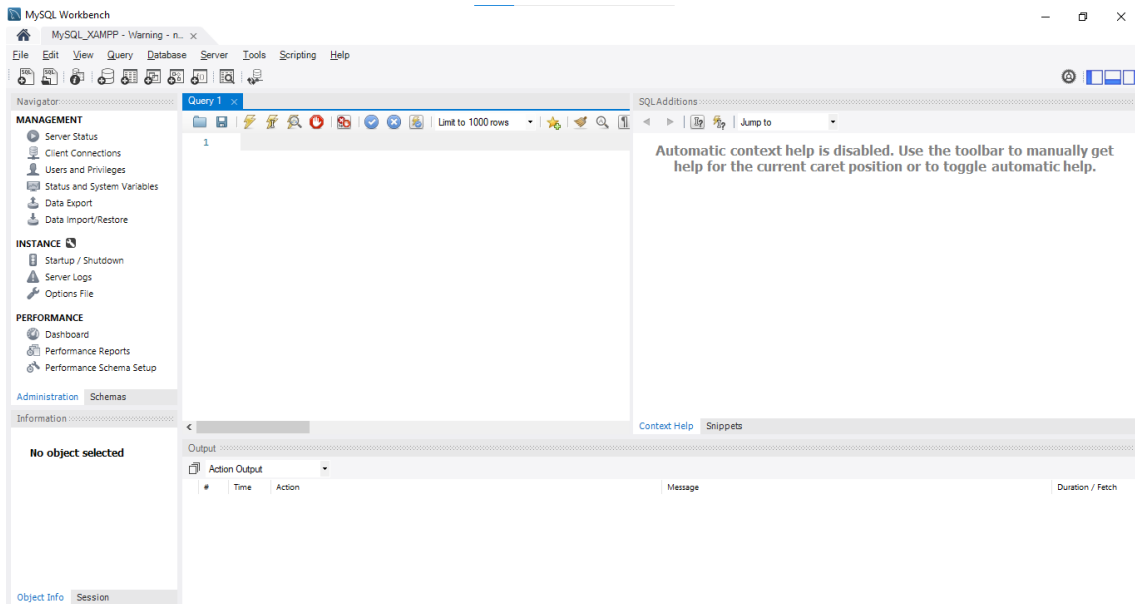
Vamos então acessar o ambiente da ferramenta Workbench pelo menu inicial:



Será apresentada a seguinte interface, selecione a conexão correspondente ao XAMPP.



**Observação:** a senha é a mesma utilizada para a interface de comando que foi definida na instalação do SGBD. Após a digitação da senha, será apresentada a seguinte interface, isso quer dizer que o Workbench está pronto para ser utilizado.



Tudo pronto!!! Vamos utilizar o contexto de uma escola que necessita gerenciar o cadastro de seus professores.

Mas antes, veja algumas informações importantes sobre esse contexto:

- 1) O professor pode ter vários telefones e lecionar em vários cursos;
- 2) Um curso pode ter vários professores;
- 3) Um telefone não pode ser cadastrado para o mesmo professor mais de uma vez, assim como um professor não pode estar vinculado mais de uma vez a um curso. Exemplos:

Matricula	Professor	Telefone
5234	CARLOS DA SILVA	(11) 3578-3456
5234	CARLOS DA SILVA	(11) 3578-3456

Matricula	Professor	Curso
5234	CARLOS DA SILVA	INFORMÁTICA
5234	CARLOS DA SILVA	INFORMÁTICA

- 4) A escola não pode ter cursos com o mesmo nome. Exemplo:

Codigo	Curso
10	INFORMATICA
11	INFORMATICA

Baseados nessas informações e no conhecimento obtido em agendas anteriores, chegamos no seguinte mapeamento:

**Curso:** codigo e descricao.

**Professor:** matricula, nome, dt\_nascimento, logradouro, numero, bairro, cidade, uf, cep e salário.

**Professor Telefones:** professor\_fone\_id, matricula e numero.

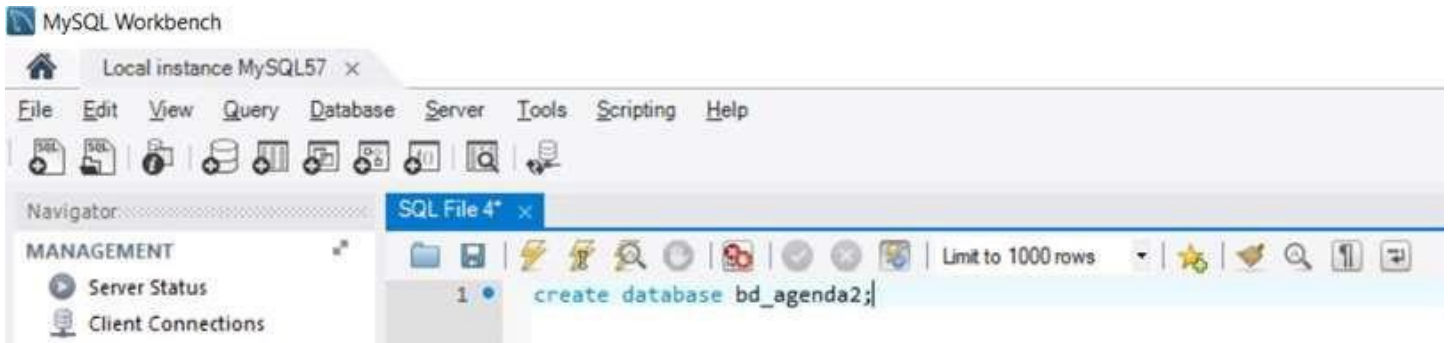
**Curso Professores:** curso\_prof\_id, codigo\_curso e matricula\_prof.

Chegou a hora de implementar o Banco de Dados, utilizando a ferramenta gráfica Workbench. Vamos utilizar pelo menos duas formas de utilização da ferramenta para cada exemplo, além de demonstrar como ficaria na interface de comando, sempre dando a opção para que você escolha entre o material escrito e o vídeo.

Primeiramente temos que criar o Banco de Dados, vamos utilizar o nome `bd_agenda2`.

1. Opção: pela janela de SQL, digite:

`create database bd_agenda2;`



Para executar os comandos digitados na Janela SQL, utilize o botão:

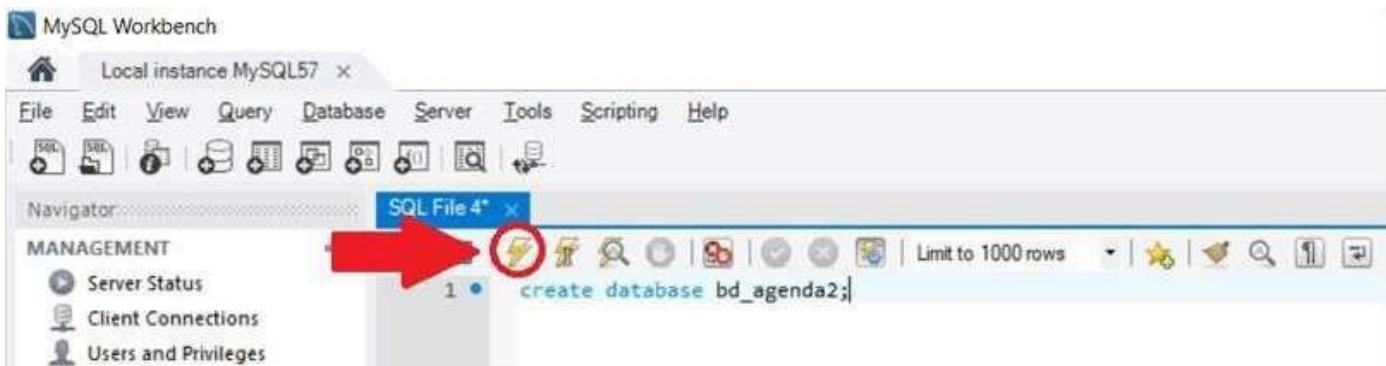
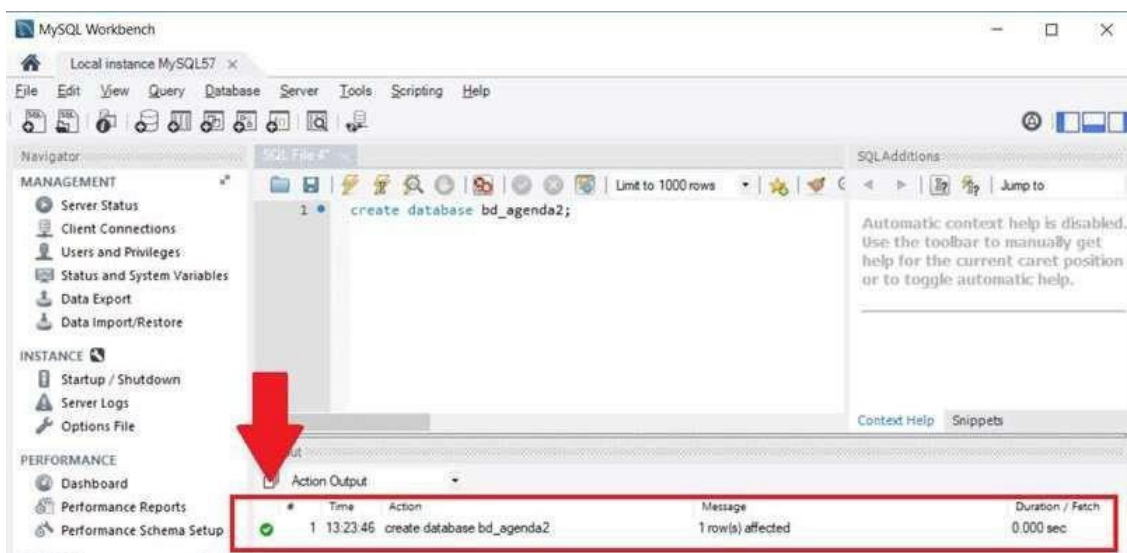


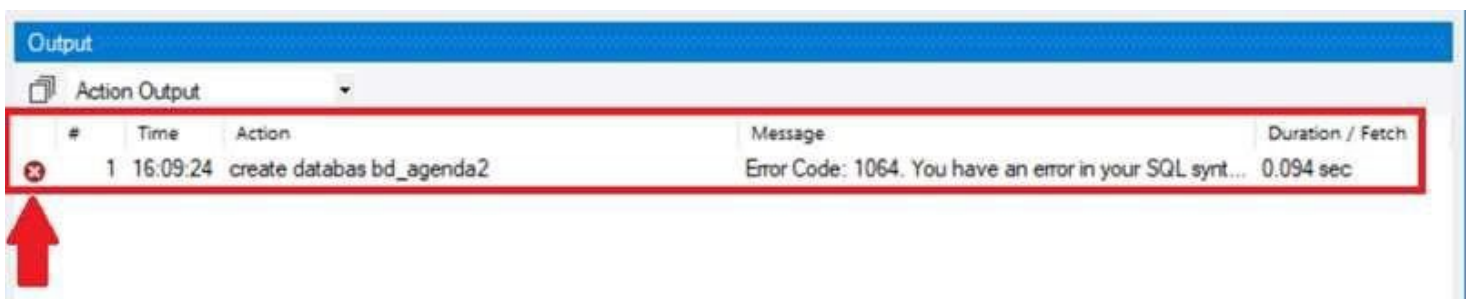
Imagem 6 – Interface Gráfica Workbench – Executar a linha de comando em SQL

Na parte inferior da tela você poderá verificar se o comando foi executado. O símbolo indica que o comando foi executado com sucesso.



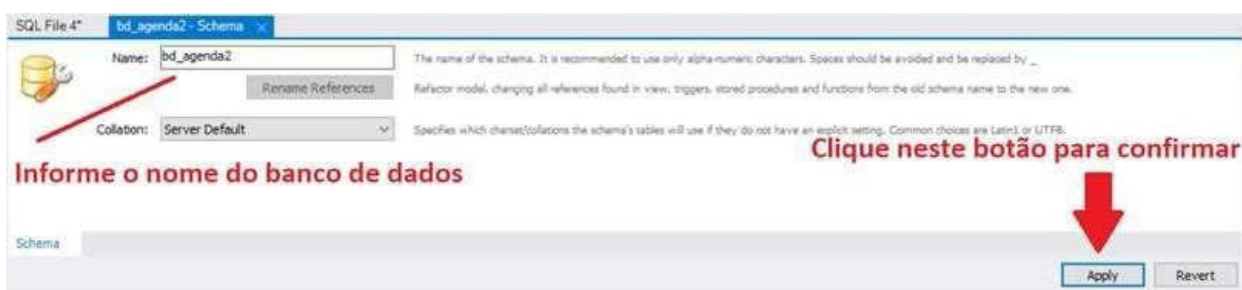
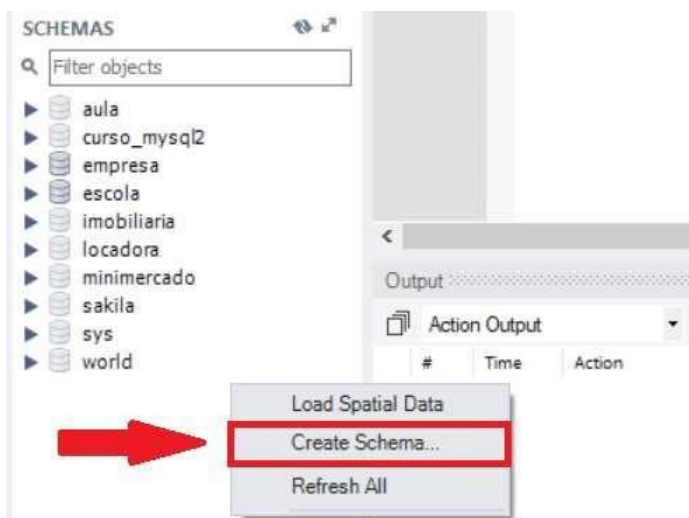


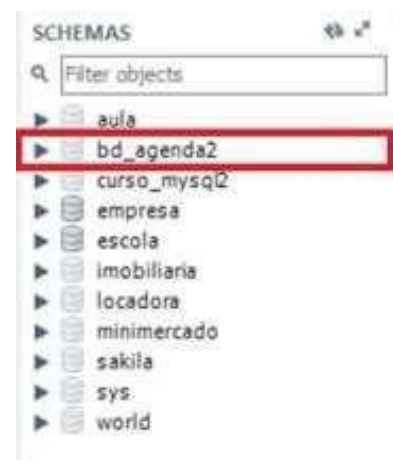
Caso haja algum problema na execução, o símbolo  é apresentado, indicando que o comando não foi executado.



**Observação:** Neste caso, o comando não foi executado porque a palavra database não foi escrita de maneira correta, ficou faltando a letra “s”.

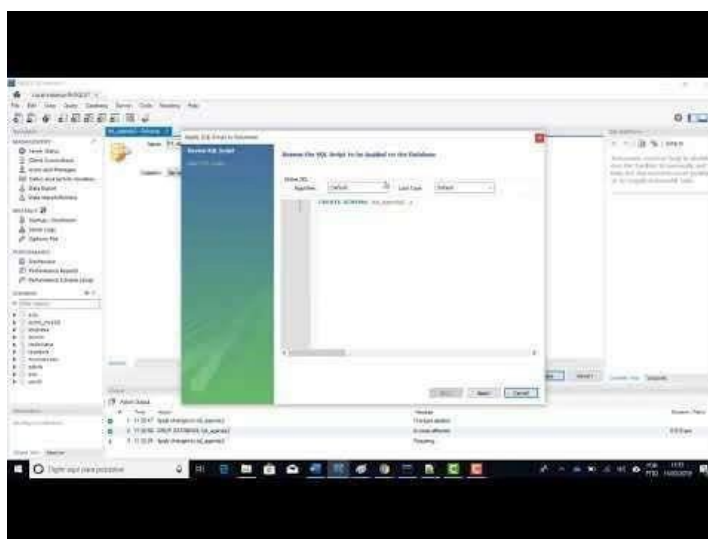
2. Opção: utilizando o botão direito do mouse, no quadro SCHEMAS:





Após a execução de todas as etapas o Banco de Dados criado é exibido no quadro SCHEMAS depois que você finalizar clicando no botão **Finish**.

### 3. Opção: assista ao vídeo por meio do [link Agenda 3 -Video2](#).




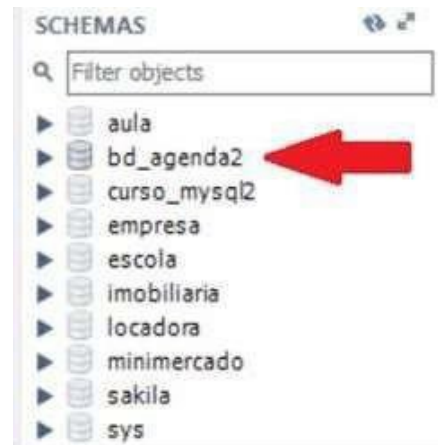
Agora chegou a vez de criarmos as Tabelas, seguindo a mesma dica da agenda anterior, vamos criar primeiramente aquelas que não possuem chave estrangeira, começando pela estrutura curso.



**Observação:** Não se esqueça de selecionar o Banco de Dados.

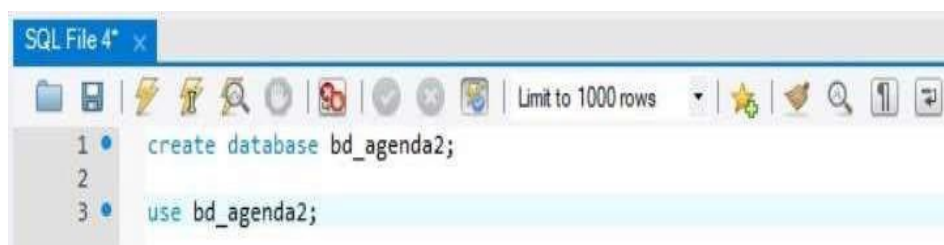
Onde as estruturas serão criadas no quadro SCHEMAS.

Com o clique duplo sobre o Banco de Dados bd\_agenda2 ou por meio da janela de  SQL, digite o comando a seguir,

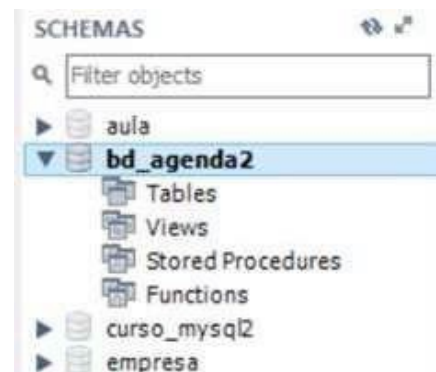


selecione e clique no botão para executar:

`use bd_agenda2;`

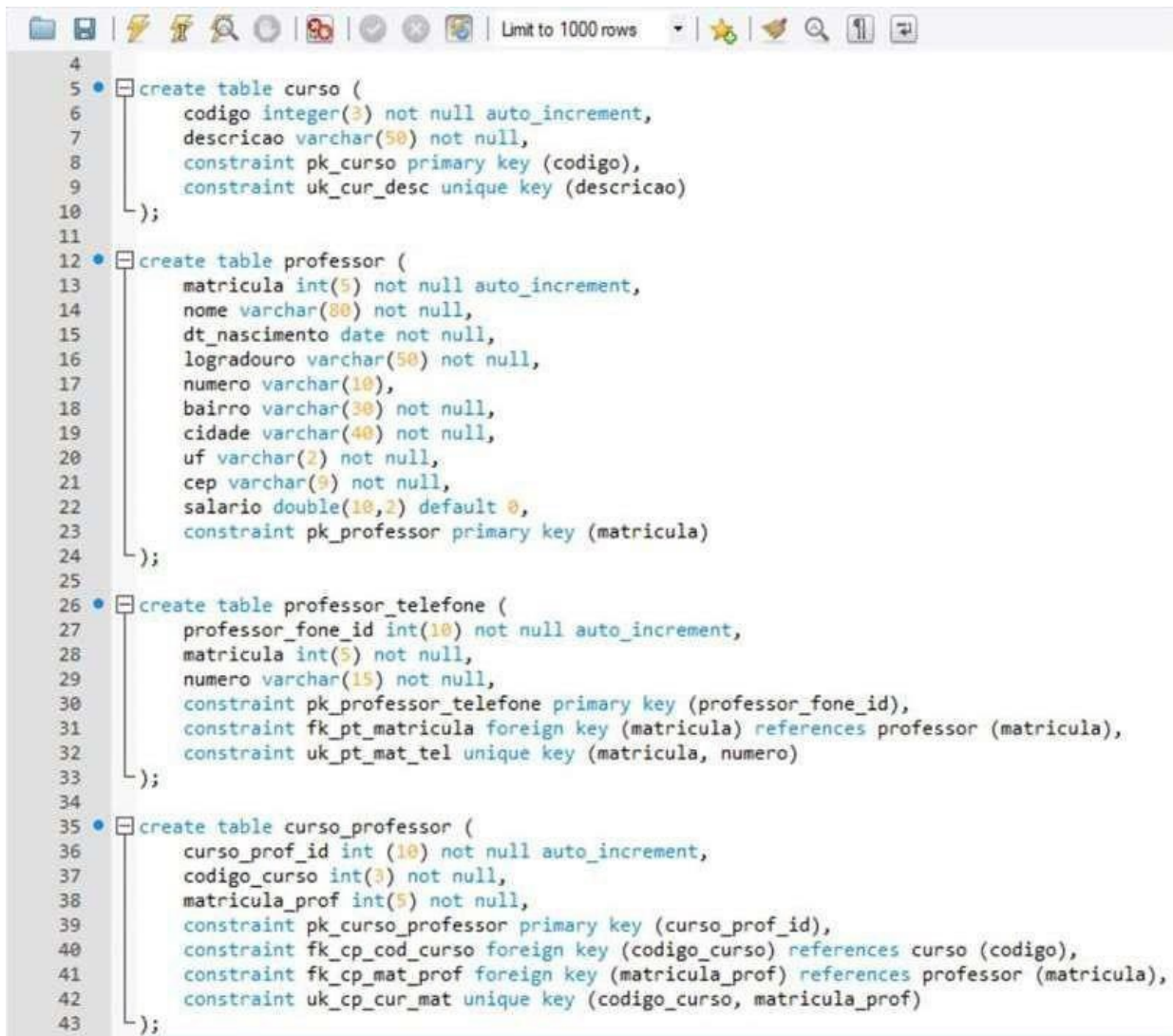


Seja pelo clique duplo ou digitando o comando na janela SQL, o nome do Banco de Dados será apresentado em **negrito** no quadro SCHEMAS conforme demonstrado:



Vamos criar as Tabelas:

**Opção 1: pela janela de SQL, digite:**



```

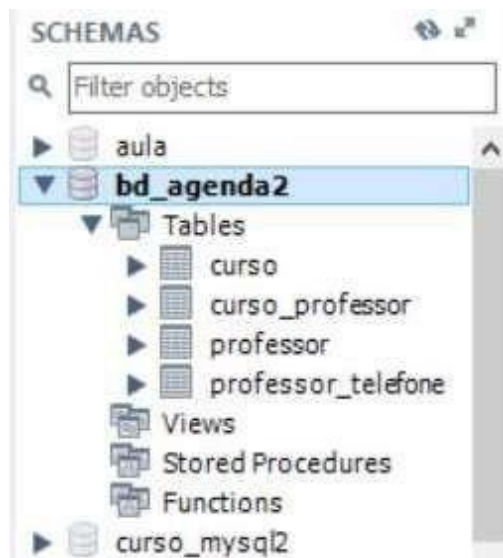
4
5 • create table curso (
6     codigo integer(3) not null auto_increment,
7     descricao varchar(50) not null,
8     constraint pk_curso primary key (codigo),
9     constraint uk_cur_desc unique key (descricao)
10 );
11
12 • create table professor (
13     matricula int(5) not null auto_increment,
14     nome varchar(80) not null,
15     dt_nascimento date not null,
16     logradouro varchar(50) not null,
17     numero varchar(10),
18     bairro varchar(30) not null,
19     cidade varchar(40) not null,
20     uf varchar(2) not null,
21     cep varchar(9) not null,
22     salario double(10,2) default 0,
23     constraint pk_professor primary key (matricula)
24 );
25
26 • create table professor_telefone (
27     professor_fone_id int(10) not null auto_increment,
28     matricula int(5) not null,
29     numero varchar(15) not null,
30     constraint pk_professor_telefone primary key (professor_fone_id),
31     constraint fk_pt_matricula foreign key (matricula) references professor (matricula),
32     constraint uk_pt_mat_tel unique key (matricula, numero)
33 );
34
35 • create table curso_professor (
36     curso_prof_id int (10) not null auto_increment,
37     codigo_curso int(3) not null,
38     matricula_prof int(5) not null,
39     constraint pk_curso_professor primary key (curso_prof_id),
40     constraint fk_cp_cod_curso foreign key (codigo_curso) references curso (codigo),
41     constraint fk_cp_mat_prof foreign key (matricula_prof) references professor (matricula),
42     constraint uk_cp_cur_mat unique key (codigo_curso, matricula_prof)
43 );
  
```

Selecione todas as linhas com a cláusula `create table` e execute clicando no botão 

Output		
Action Output		
#	Time	Action
✓ 6	21:39:52	create table curso (codigo integer(3) not null auto_increment, descricao varchar(50) not null, constraint pk_curso primary key
✓ 7	21:39:53	create table professor (matricula int(5) not null auto_increment, nome varchar(80) not null, dt_nascimento date not null,
✓ 8	21:39:53	create table professor_telefone (professor_fone_id int(10) not null auto_increment, matricula int(5) not null, numero varchar
✓ 9	21:39:53	create table curso_professor (curso_prof_id int (10) not null auto_increment, codigo_curso int(3) not null, matricula_prof i

Imagem 17 – Interface Gráfica Workbench – Janela de Output confirmando a execução dos comandos.

Após a execução clique no botão do quadro SCHEMAS para que as Tabelas criadas sejam exibidas:

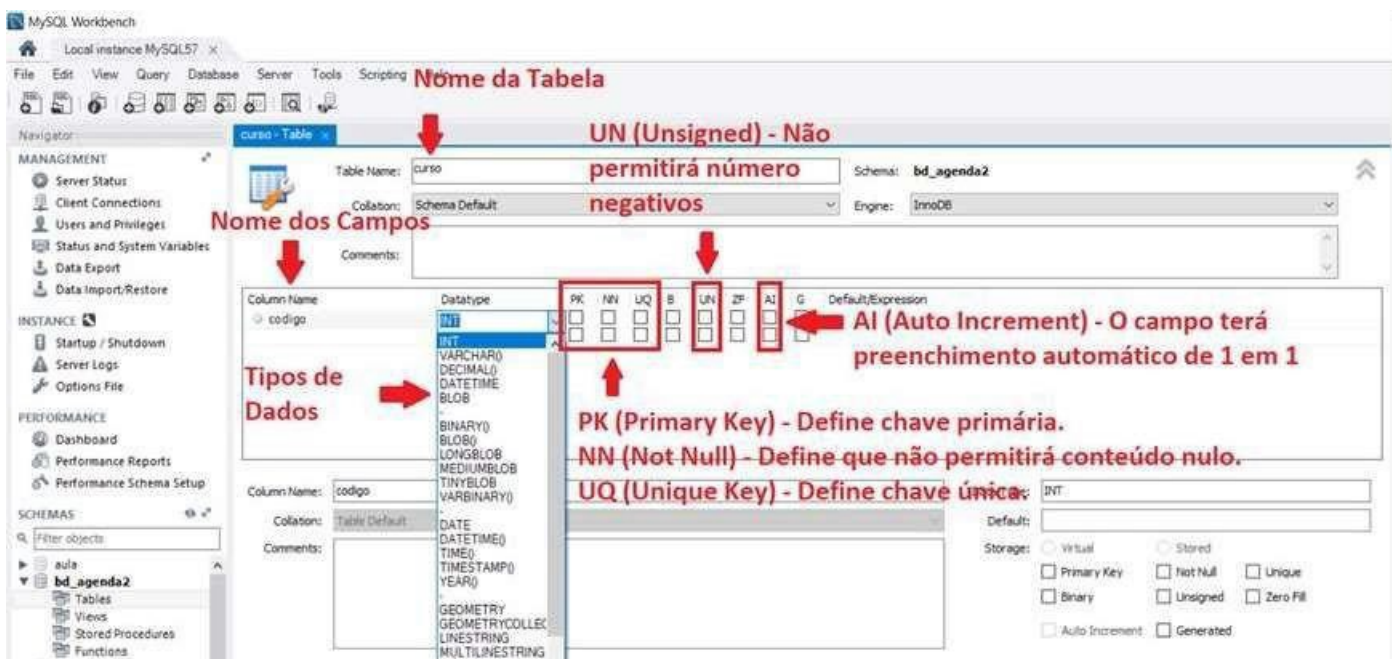
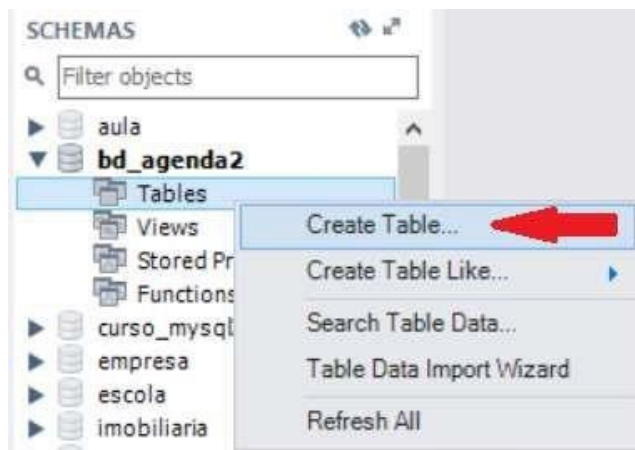


**Observação:** As chaves únicas utilizadas na criação das Tabelas curso, professor\_telefone e curso\_professor, têm o objetivo de cumprir os seguintes requisitos:

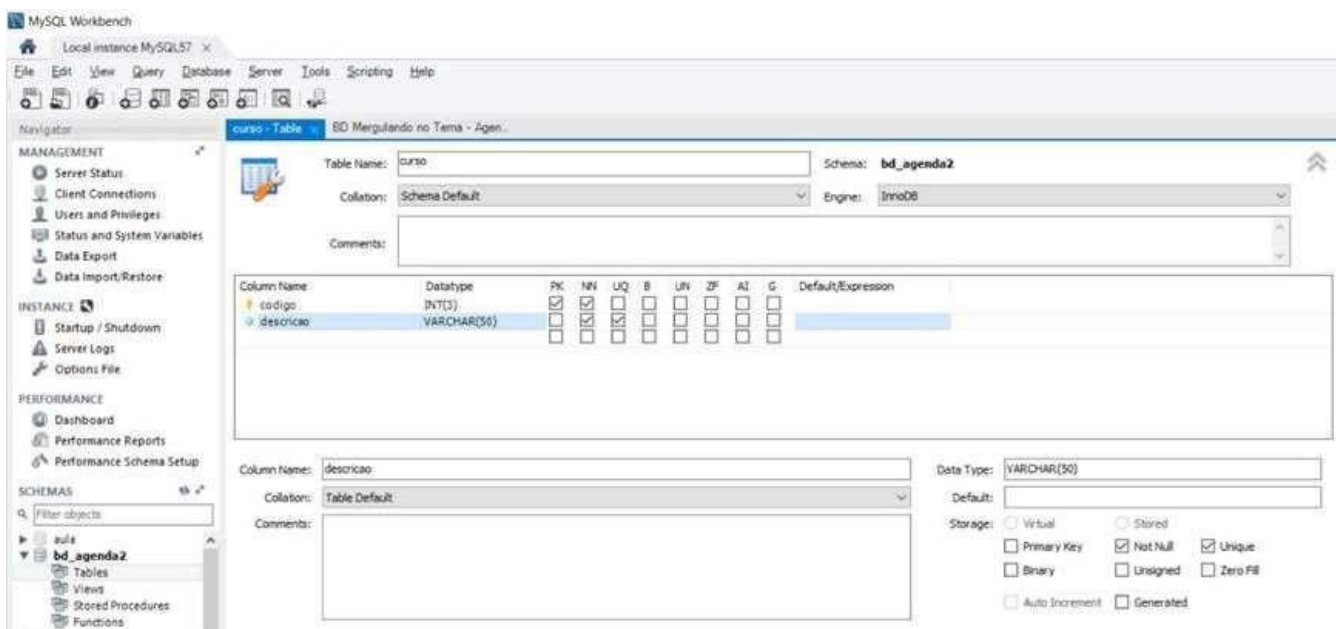
- Um telefone não pode ser cadastrado para o mesmo professor mais de uma vez, assim como um professor não pode estar vinculado mais de uma vez a um curso;
  - A escola não pode ter cursos com o mesmo nome.
- Já as chaves estrangeiras na criação das Tabelas professor\_telefone e curso\_professor, são utilizadas para cumprir os seguintes requisitos: - O professor pode ter vários telefones e lecionar em vários cursos; - Um curso pode ter vários professores.

#### Opção 2: utilizando o botão direito do mouse, no quadro SCHEMAS:

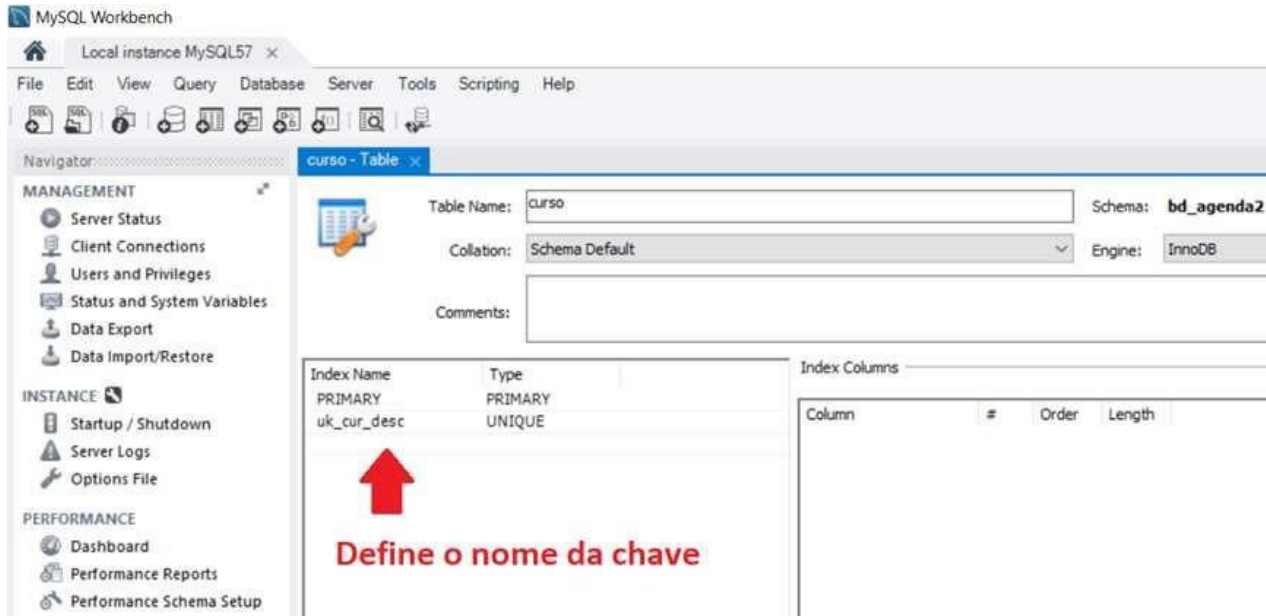
Utilizando o botão direito do mouse clique em  **Tables** do Banco de Dados selecionado bd\_agenda\_2, no quadro SCHEMAS:



Vamos utilizar a interface para criar a Tabela curso baseado no mapeamento realizado.

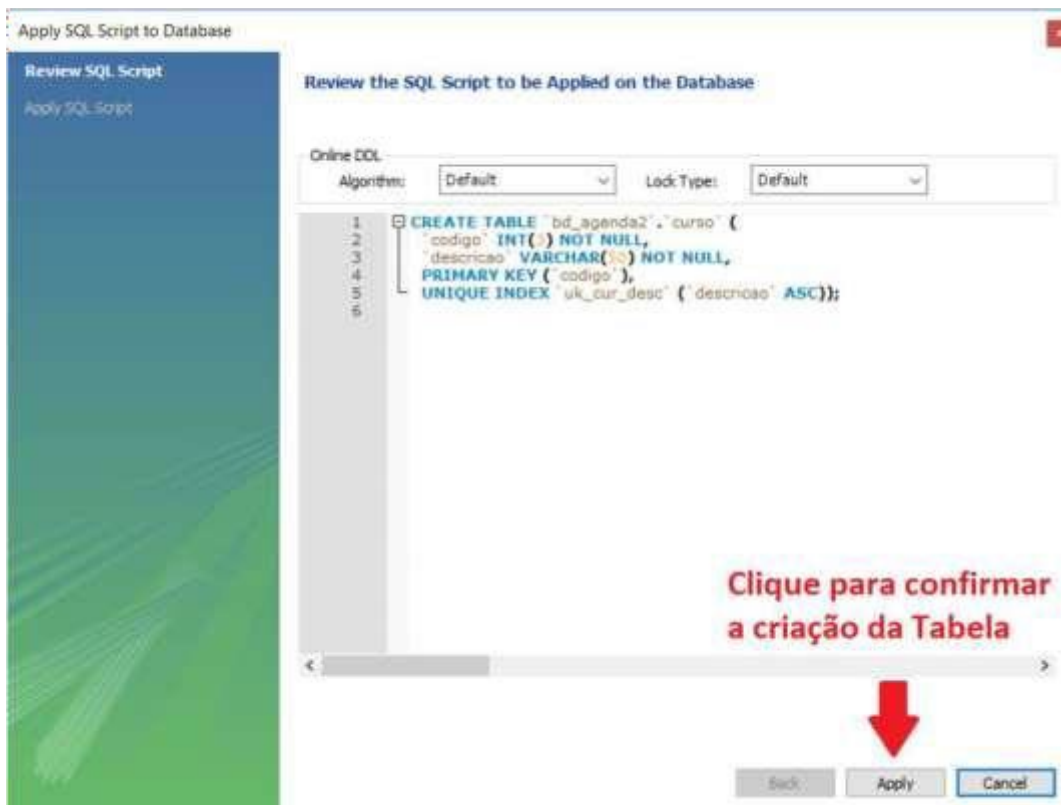


Após a definição dos campos e suas restrições, vamos nomear a **chave única-chave primária**. O MySQL, por padrão, apresenta essa chave como PRIMARY, mas o importante é que você aprenda a nomeá-la em outros SGBDs. A sintaxe do comando é a mesma, o que você define é o apresentado como, por exemplo, no SGBD Oracle.



A Tabela curso não possui chave estrangeira, nesse caso, podemos finalizar o processo clicando no botão

Apply





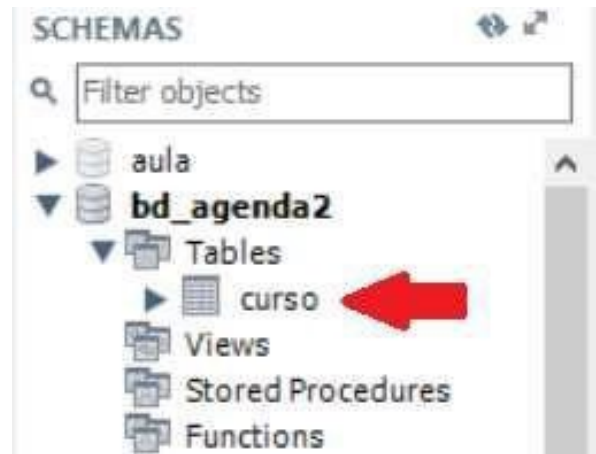
Observe que mesmo utilizando uma interface gráfica, assim como na criação do Banco de Dados, no final do processo é gerado o comando SQL necessário para criação da Tabela, conforme Imagem 23. O SQL gerado é um pouco diferente do que estamos utilizando, mas não se preocupe, o mais importante é o entendimento da sintaxe completa, uma vez que você se familiariza com a Linguagem SQL, cada vez mais você utilizará seus

recursos baseados no SGBD escolhido para seus projetos.

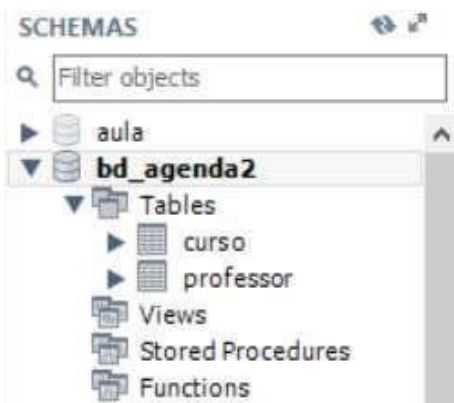
Resumir os comandos fazem parte desse processo.

Após a execução será apresentada a interface de finalização, conforme Imagem 1. A tabela curso será apresentada no quadro SCHEMAS, abaixo de **Tables** na árvore hierárquica do Banco de Dados bd\_agenda2.

Finalize clicando no botão **Finish**



Opção 3: Assista ao [vídeo por meio do link Agenda 3 - Vídeo 3.](#)



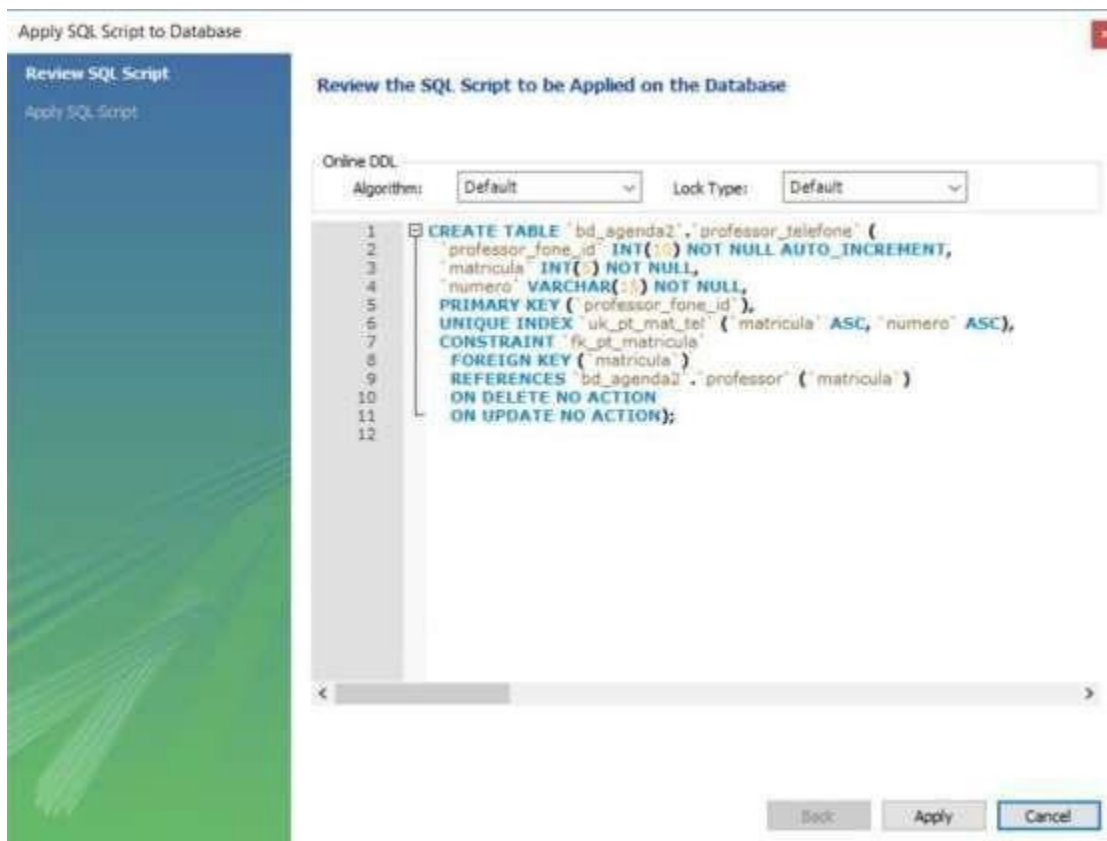
Para criação da Tabela professor, o mesmo processo utilizado para criação da Tabela curso é aplicado, com isso o quadro SCHEMAS apresentará as duas estruturas.

Vamos agora criar a tabela **professor\_telefone**, que possui **chaves estrangeira e única**. O processo é o mesmo utilizado para criação das duas estruturas anteriores, **curso** e **professor**. Vamos adicionar somente o processo de criação da chave estrangeira pela interface gráfica, utilizando a aba Foreign Keys ou assistindo ao vídeo por meio do link [Agenda 3 - Vídeo 4.mp4](#).



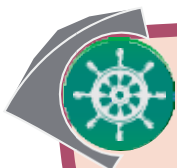
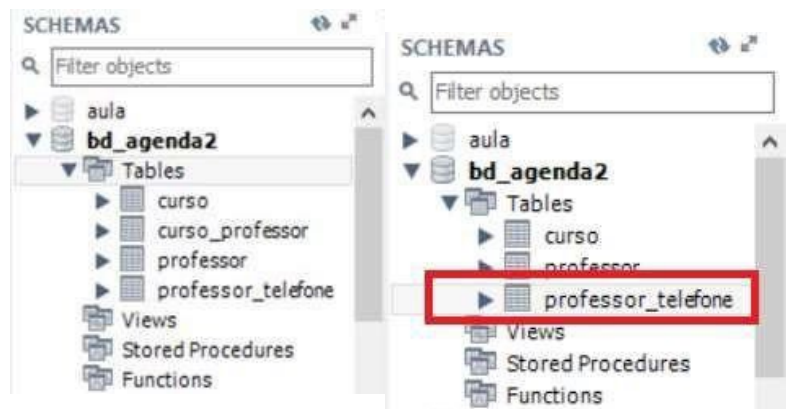
Após definir as características da chave estrangeira, finalize o processo clicando no botão

**Apply**



Após a confirmação, a Tabela **professor\_telefone** será apresentada no quadro SCHEMAS.

Para criar a tabela **curso\_professor**, foi utilizado o mesmo processo, todas as estruturas criadas serão apresentadas na árvore hierárquica no Banco de Dados **bd\_agenda2**, com isso terminamos a criação de todas as estruturas do projeto da escola.



## VOCÊ NO COMANDO

*Vamos utilizar no minimercado do Adriano, o Projeto do Banco de Dados do Sistema que já está pronto para ser implementado. Esse contexto tem um desafio, a implementação de valores padrões. Como será desenvolvê-lo em uma interface gráfica? Utilize o Workbench para auxiliá-lo nessa tarefa.*

PRODUTO							
nome	tipo de dados	tamanho	obrigatório	único	chave primária	chave estrangeira	valor default
codigo_produto	INT	5	sim	sim	sim	não	não
descricao	VARCHAR	40	sim	não	não	não	não
quantidade_estoque	DOUBLE	10,1	sim	não	não	não	0

CLIENTE							
nome	tipo de dados	tamanho	obrigatório	único	chave primária	chave estrangeira	valor default
cpf_cliente	VARCHAR	11	sim	sim	sim	não	não
nome	VARCHAR	80	sim	não	não	não	não
endereco	VARCHAR	80	sim	não	não	não	não

COMPRA_PRODUTO							
nome	tipo de dados	tamanho	obrigatório	único	chave primária	chave estrangeira	valor default
id_compra_produto	INT	11	sim	sim	sim	não	não
codigo_compra	INT	11	sim	não	não	sim	não
codigo_produto	INT	5	sim	não	não	sim	não
quantidade	DOUBLE	10,1	sim	não	não	não	1
preco	DOUBLE	10,2	sim	não	não	não	não

COMPRA							
nome	tipo de dados	tamanho	obrigatório	único	chave primária	chave estrangeira	valor default
codigo_compra	INT	11	sim	sim	sim	não	não
data	DATE	10	sim	não	não	não	não
cpf_cliente	VARCHAR	11	não	não	não	sim	não

Tudo certo?

Você conseguiu?

Tenho certeza que sim!!!! Vamos juntos conferir!!!

Para desenvolver as estruturas **produto** e **compra\_produto** além do processo normal, foi necessário definir para os campos **quantidade\_estoque** e **quantidade**, dessas estruturas, os valores padrões **0** e **1** respectivamente. Utilizando a janela de SQL, foram necessários os seguintes comandos:

```

SQL Minimercado Adriano x
Limit to 1000 rows

1 • create database minimercado;
2
3 • use minimercado;
4
5 • create table produto (
6     codigo_produto int(5) not null auto_increment,
7     descricao varchar(40) not null,
8     quantidade_estoque double(10,1) not null default 0,
9     constraint pk_produto primary key (codigo_produto)
10 );
11
12 • create table cliente (
13     cpf_cliente varchar(11) not null,
14     nome varchar(80) not null,
15     endereco varchar(80) not null,
16     constraint pk_cliente primary key (cpf_cliente)
17 );
18
19 • create table compra (
20     codigo_compra int(11) not null auto_increment,
21     data date not null,
22     cpf_cliente varchar(11) not null,
23     constraint pk_compra primary key (codigo_compra),
24     constraint fk_compra_cpf_cliente foreign key (cpf_cliente)
25         references cliente (cpf_cliente)
26 );
27
28 • create table compra_produto (
29     id_compra_produto int(11) not null auto_increment,
30     codigo_compra int(11) not null,
31     codigo_produto int(5) not null,
32     quantidade double(10,1) not null default 1,
33     preco double(10,2) not null,
34     constraint pk_compra_produto primary key (id_compra_produto),
35     constraint fk_compprod_codigo_compra foreign key (codigo_compra)
36         references compra (codigo_compra),
37     constraint fk_compprod_codigo_produto foreign key (codigo_produto)
38         references produto (codigo_produto)
39 );

```

**Observação:** Para cumprir o desafio foi utilizado a cláusula `default`.

No processo de criação das estruturas por meio da interface do Workbench foram necessários definir os valores para a cláusula `default`.

produto - Table

Table Name:  Schema: **minimercado**

Collation:  Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
codigo_produto	INT(5)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
descricao	VARCHAR(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
quantidade_estoque	DOUBLE(10,1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Define o valor padrão do campo caso o mesmo não for informado quando da inclusão do registro.

Column Name:  Collation:  Comments:

Data Type:  Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

compra\_produto - Table

Table Name:  Schema: **minimercado**

Collation:  Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_compra_produto	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
codigo_compra	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
codigo_produto	INT(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
quantidade	DOUBLE(10,1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
preco	DOUBLE(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:  Collation:  Comments:

Data Type:  Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

É isso aí!!! Vamos agora finalizar essa agenda colocando a mão na massa.