ânima EDUCAÇÃO

UNISUL
UNIVERSIDADE DO SUL DE SANTA CATARINA

MODELAGEM DE SOFTWARE

Prof. Saulo Popov Zambiasi

Prof. Richard Henrique de Souza

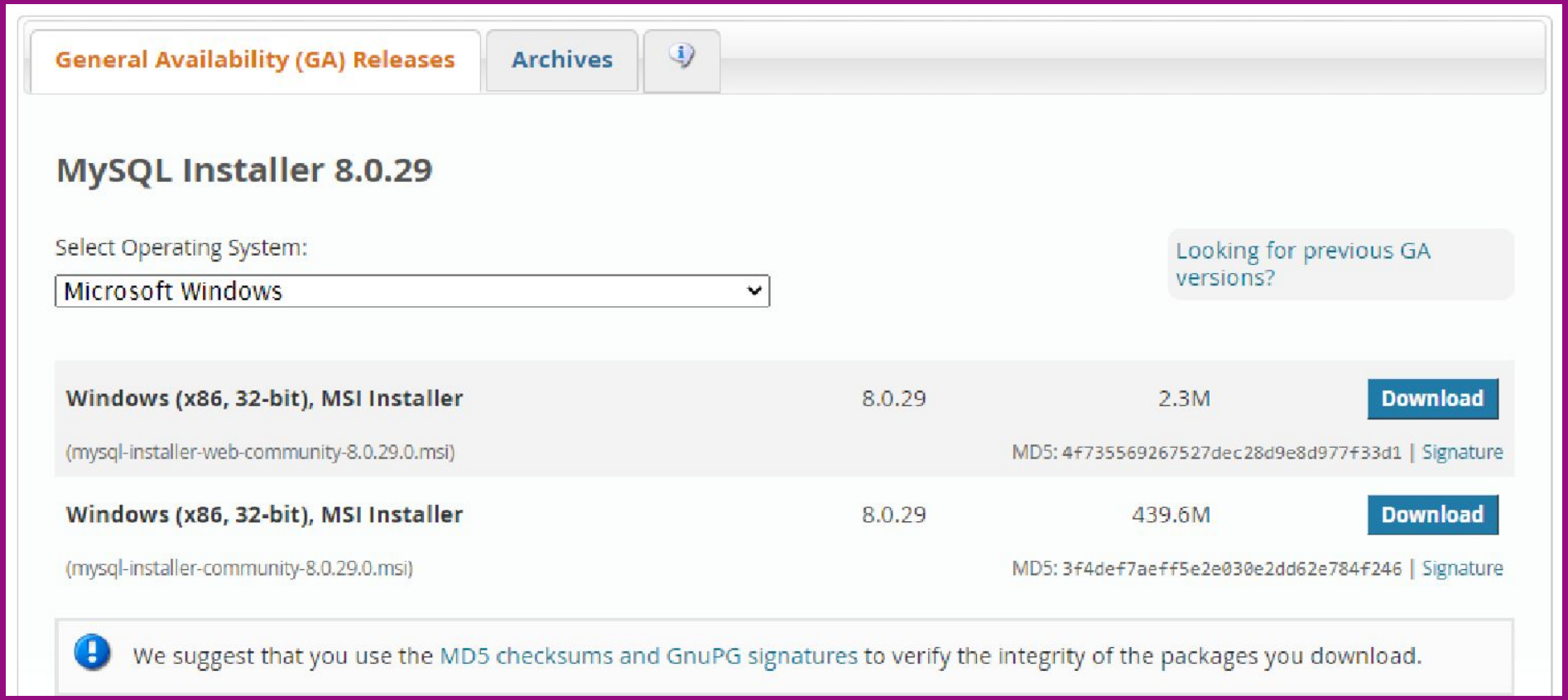
Prof. Ricardo Ribeiro Assink

Prof. Edson Lessa

Vamos instalar as ferramentas

- MySql
  - https://dev.mysql.com/downloads/installer/
    - mysql-installer-community-8.0.29.0
- Workbenach
  - https://www.mysql.com/products/workbench/

Faça o download do MySQL



Faça o download do Workbench

General Availability (GA) Releases | Archives

MySQL Workbench 8.0.29

Select Operating System:

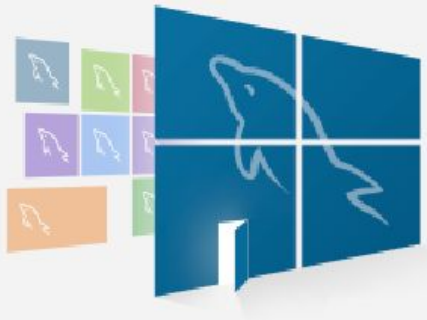
Microsoft Windows

Recommended Download:

MySQL Installer
for Windows

**All MySQL Products. For All Windows Platforms.
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

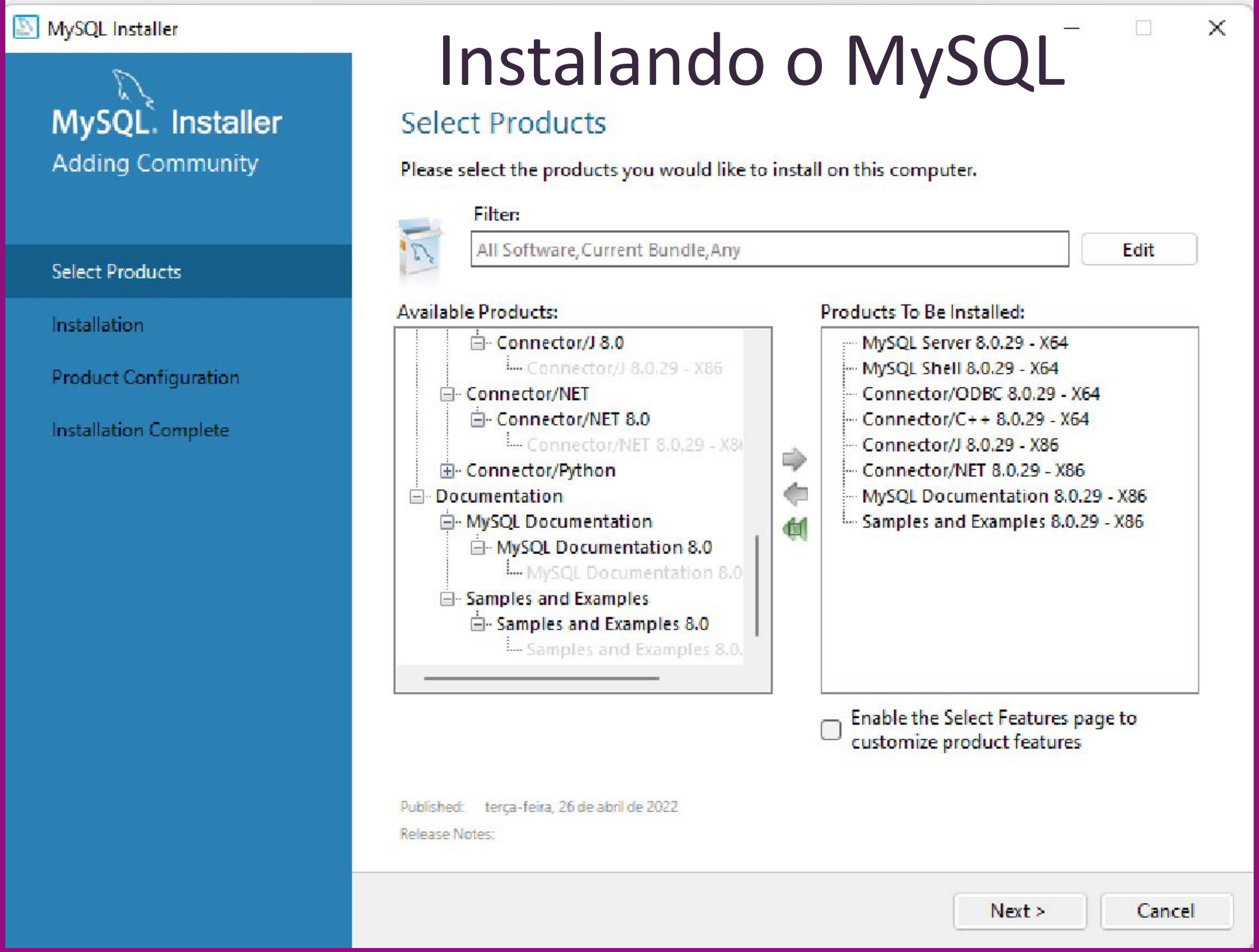


Windows (x86, 32 & 64-bit), MySQL Installer MSI — Go to Download Page >

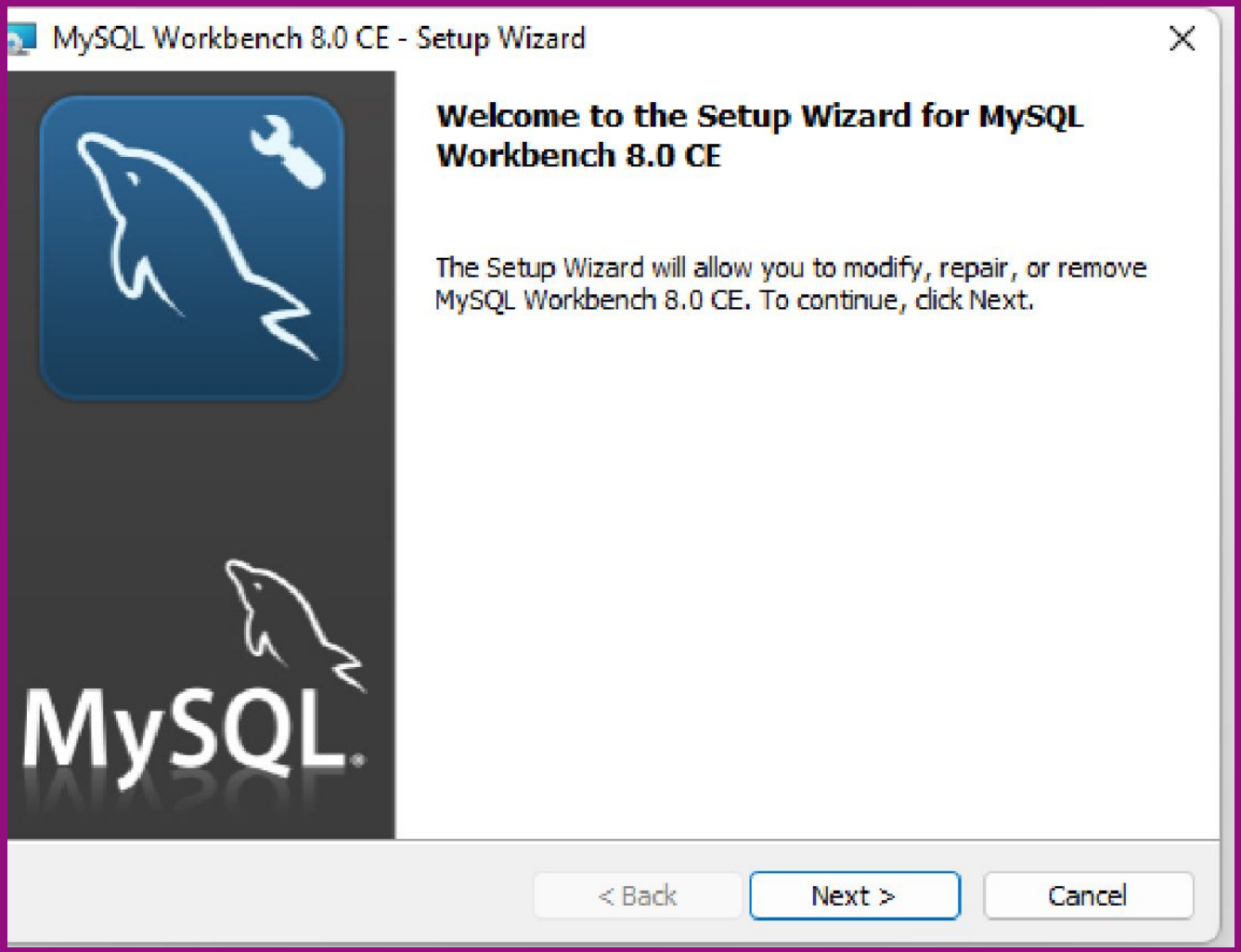
Other Downloads:

Windows (x86, 64-bit), MSI Installer	8.0.29	42.9M	Download
(mysql-workbench-community-8.0.29-winx64.msi)		MD5: 2c40a1e64dd8be34c91501749961dd9c | Signature	

Instalando o MySQL

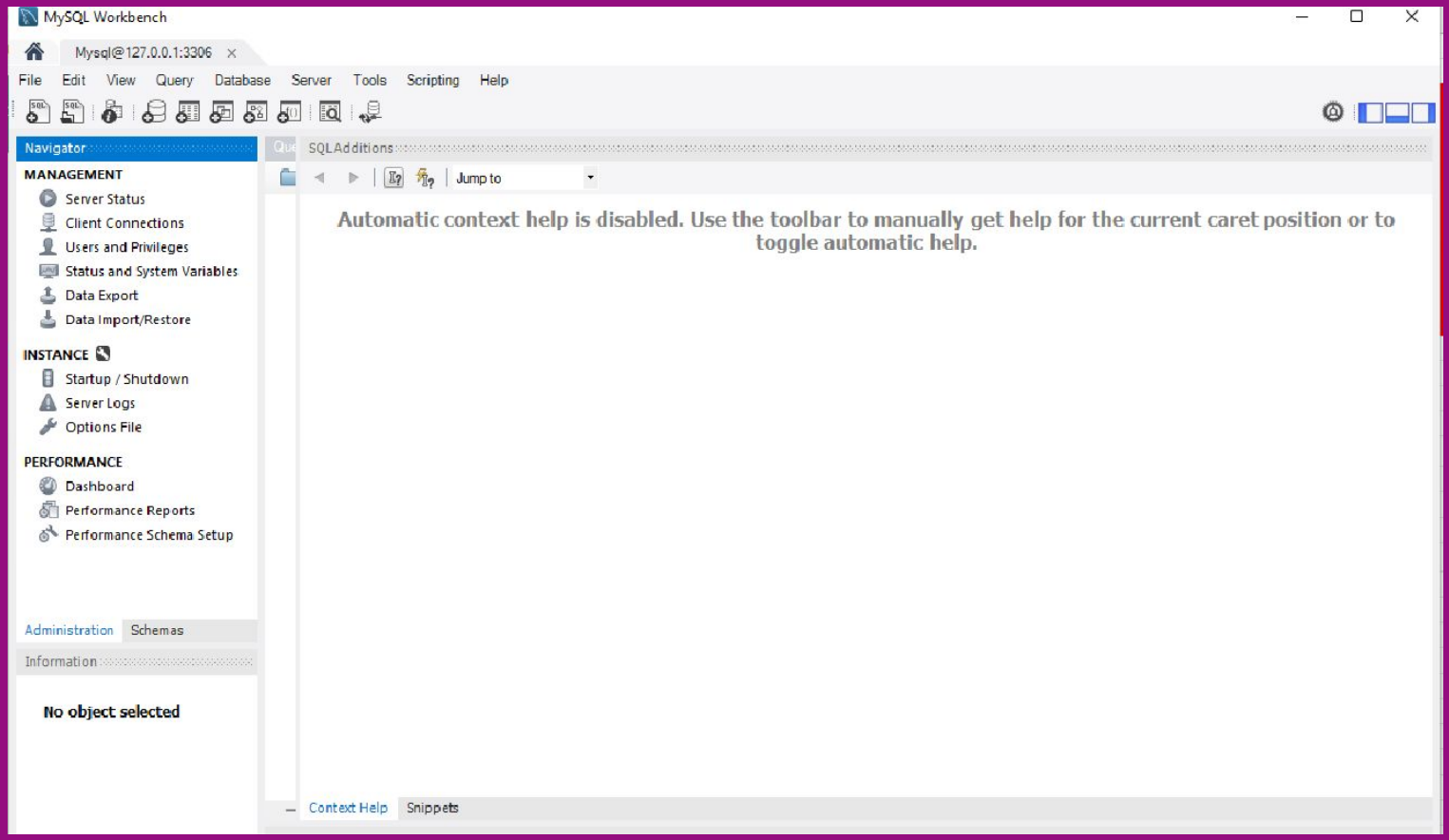


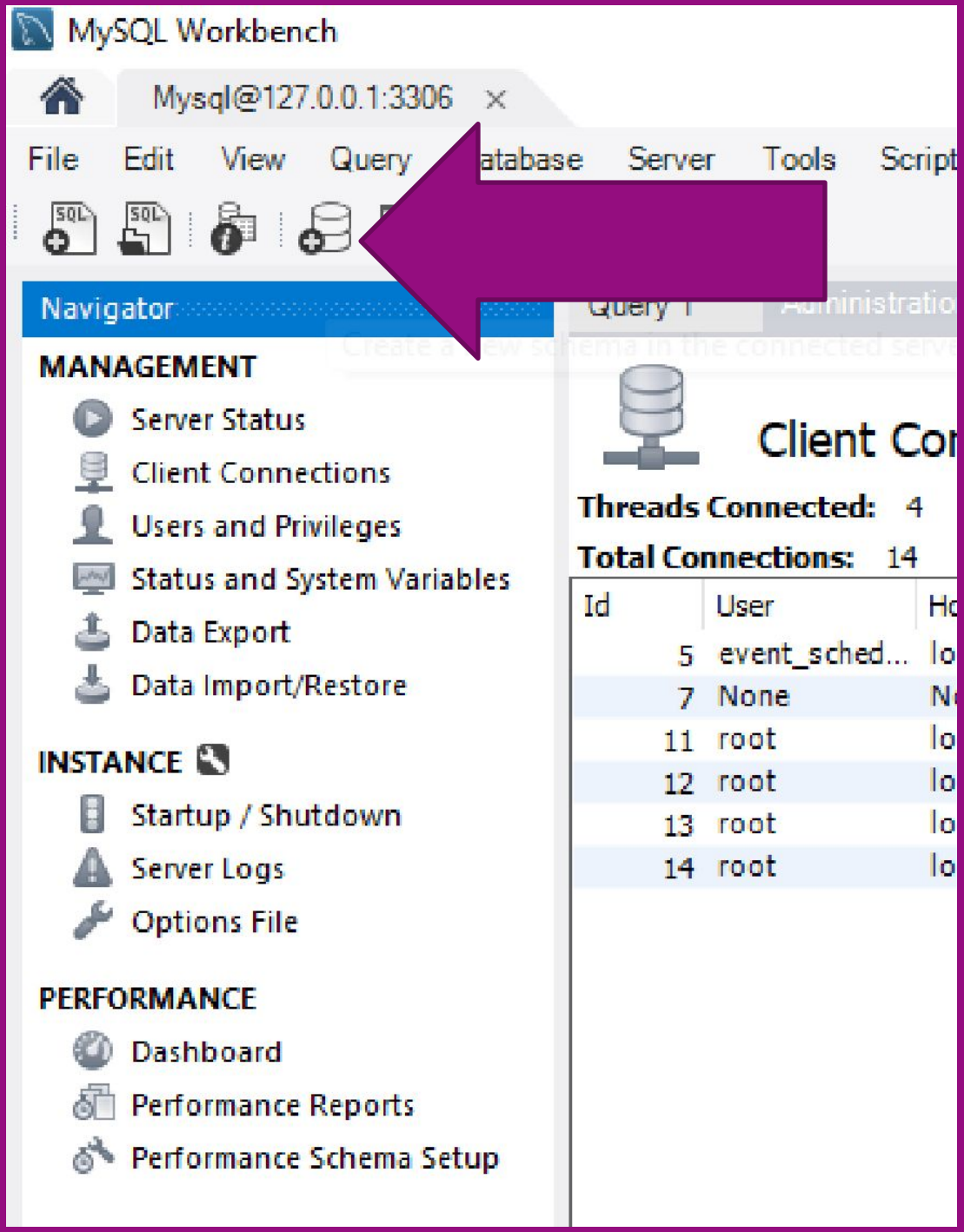
Instale o Workbench



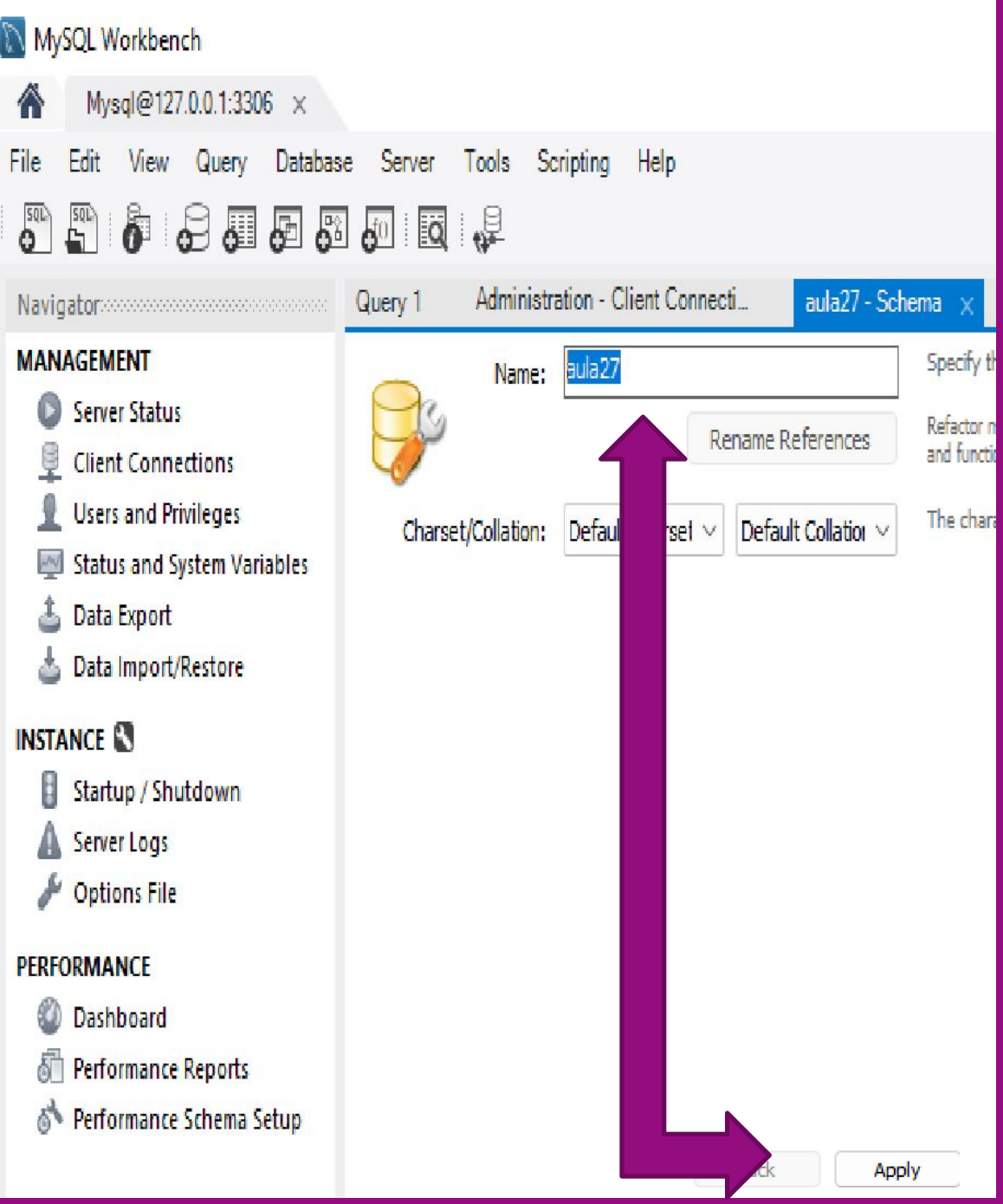
Workbench

Depois de tudo instalado, abra o workbench





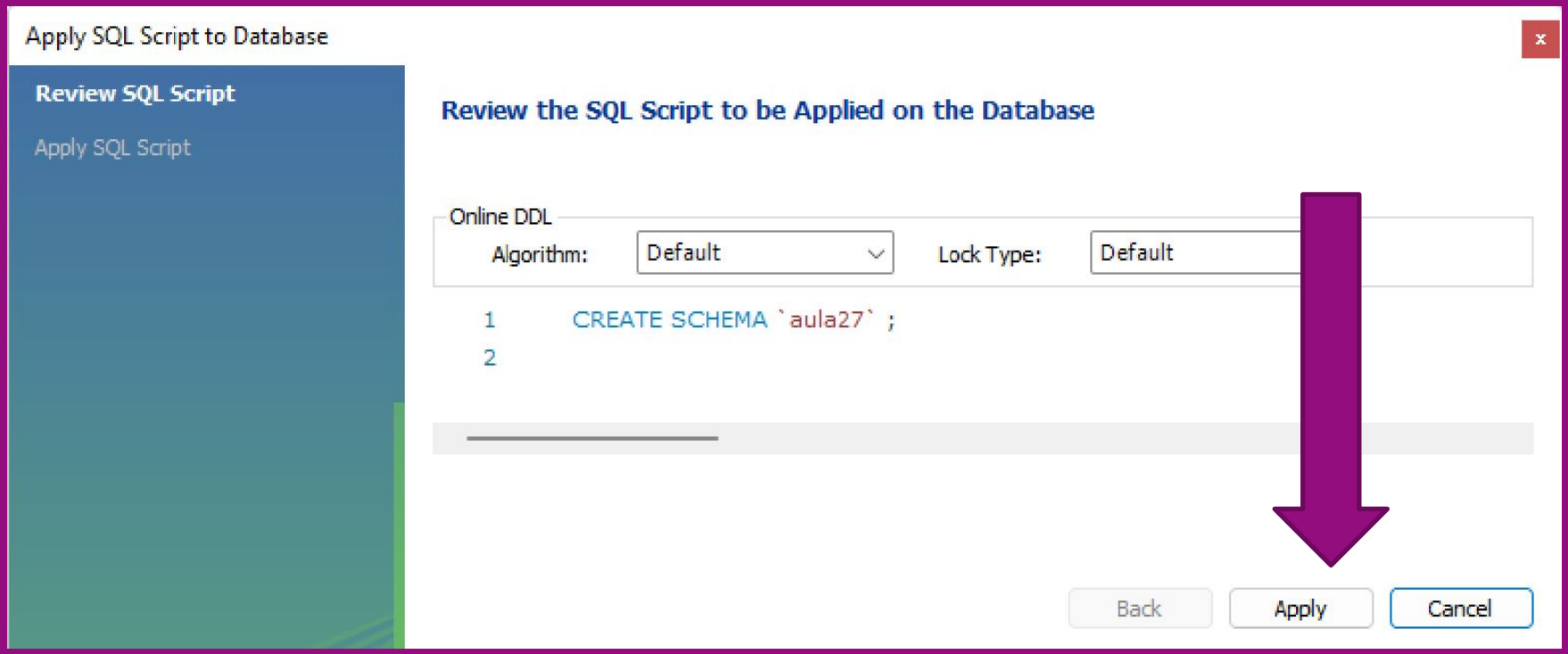
- Crie um Schema



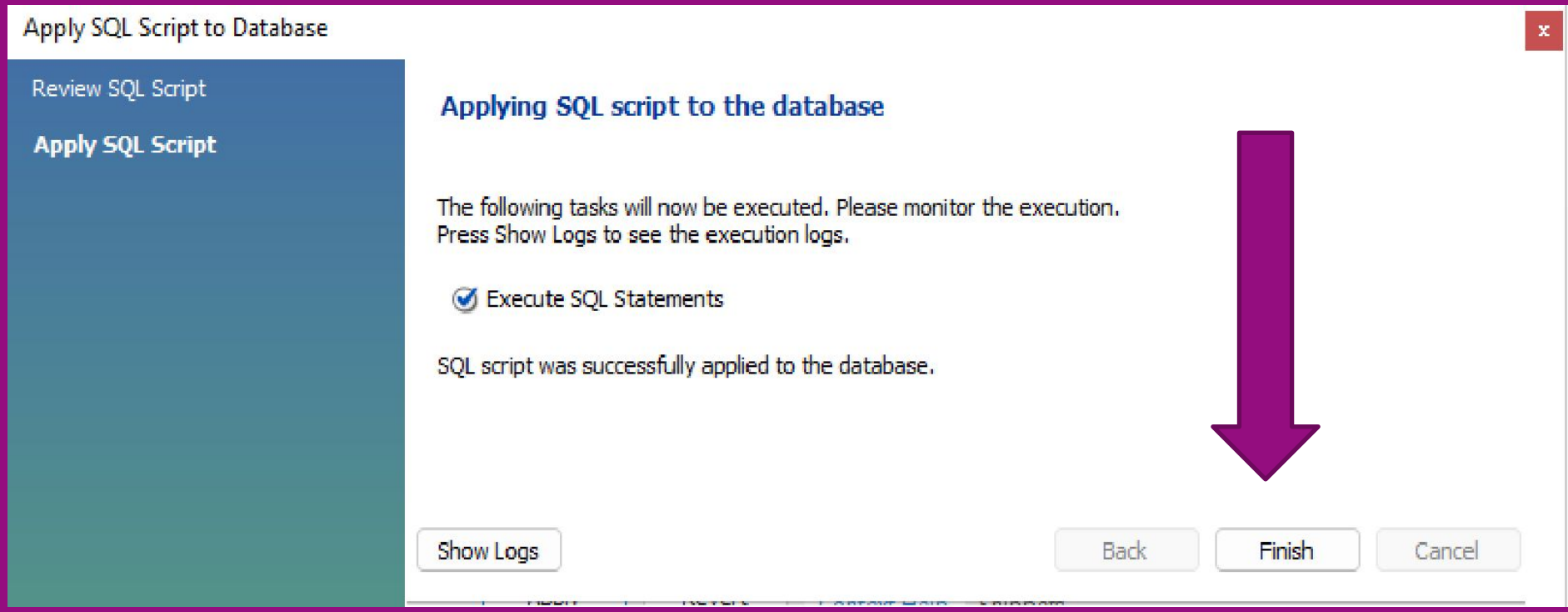
- Indique um nome
- Clique em Apply
- Exemplo aula27

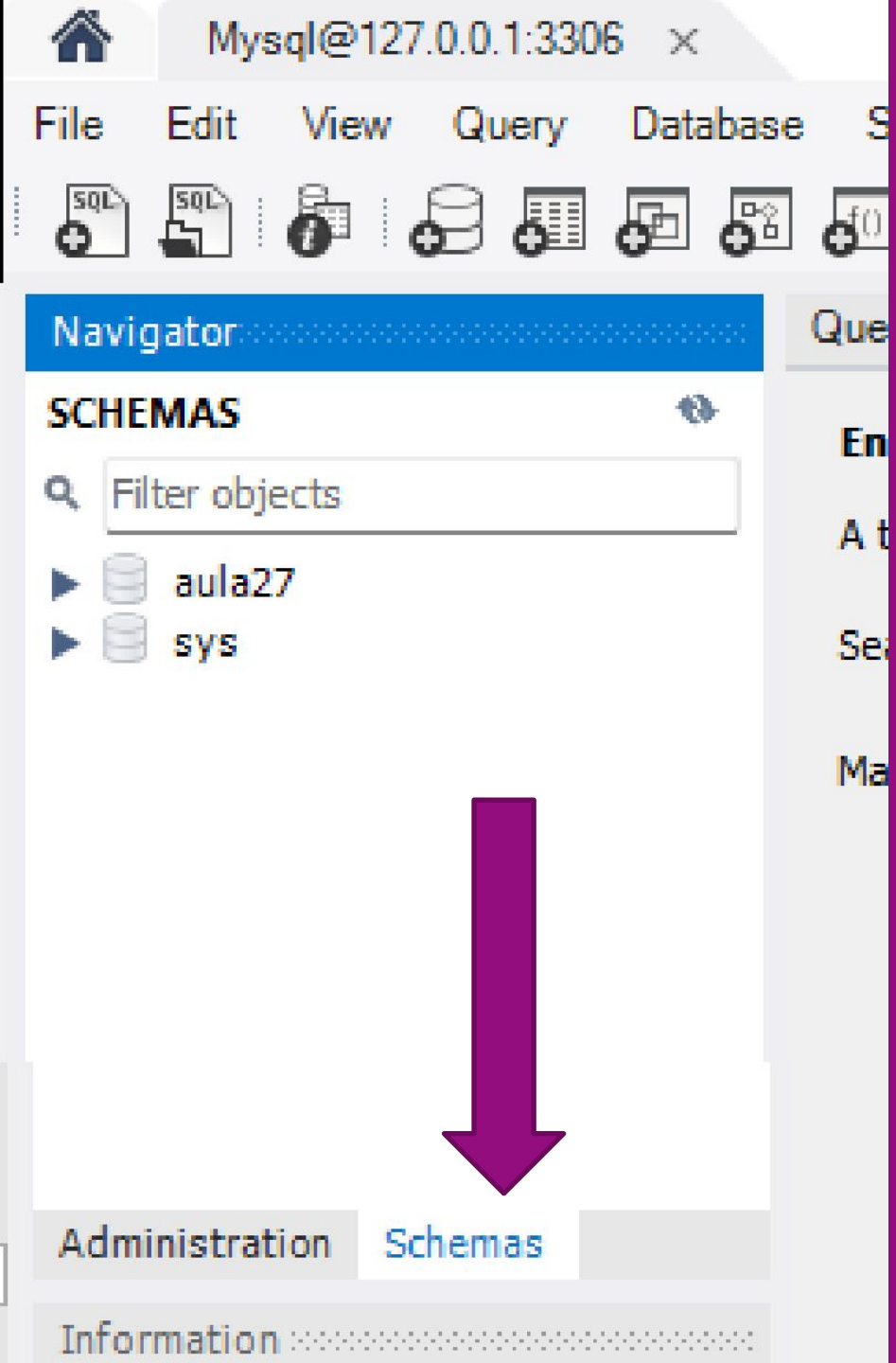
CREATE SCHEMA

- CREATE SCHEMA `aula27` ;

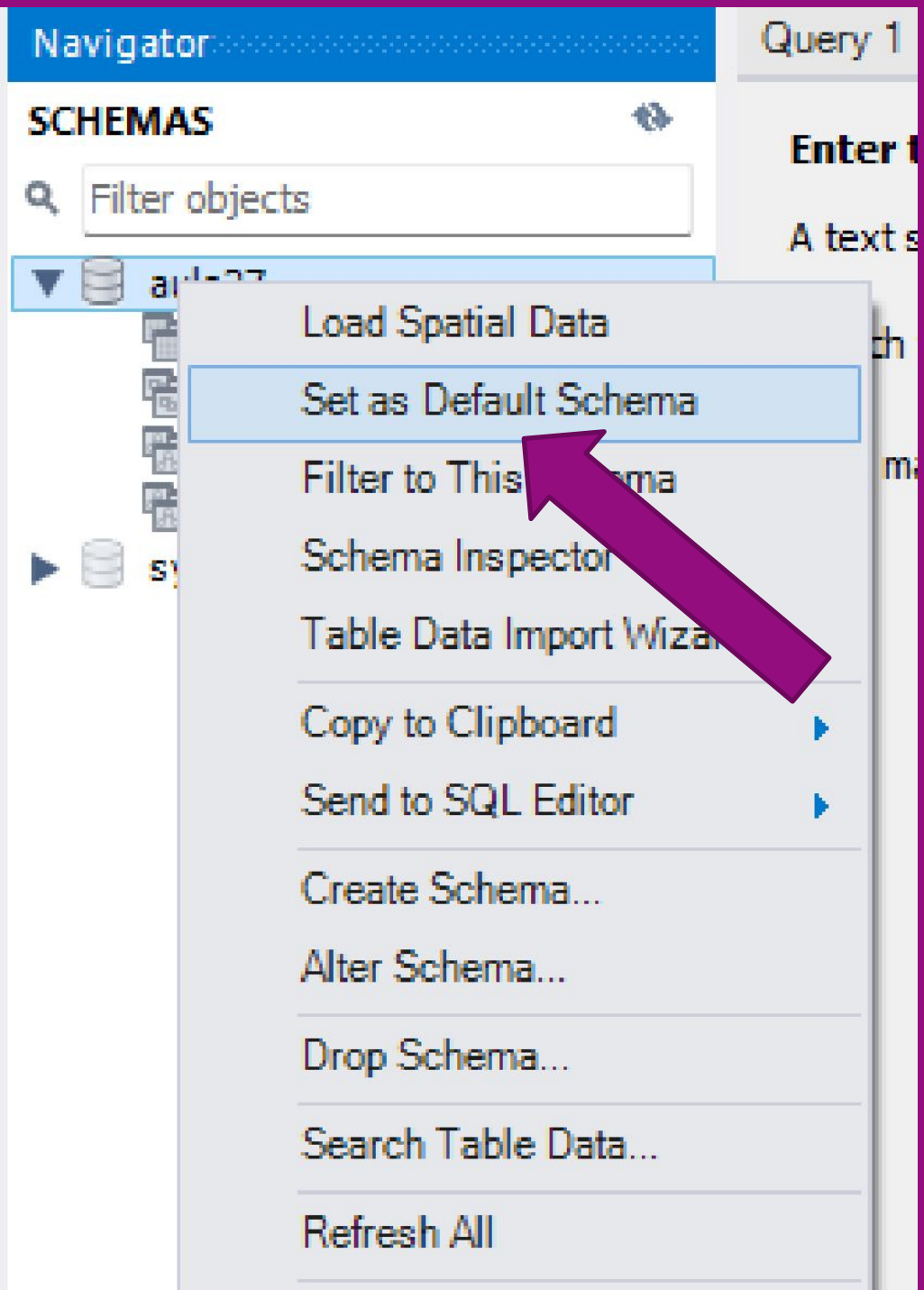


Finalise


Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Applying SQL script to the database
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.
Execute SQL Statements
SQL script was successfully applied to the database.
Show Logs
Back
Finish
Cancel

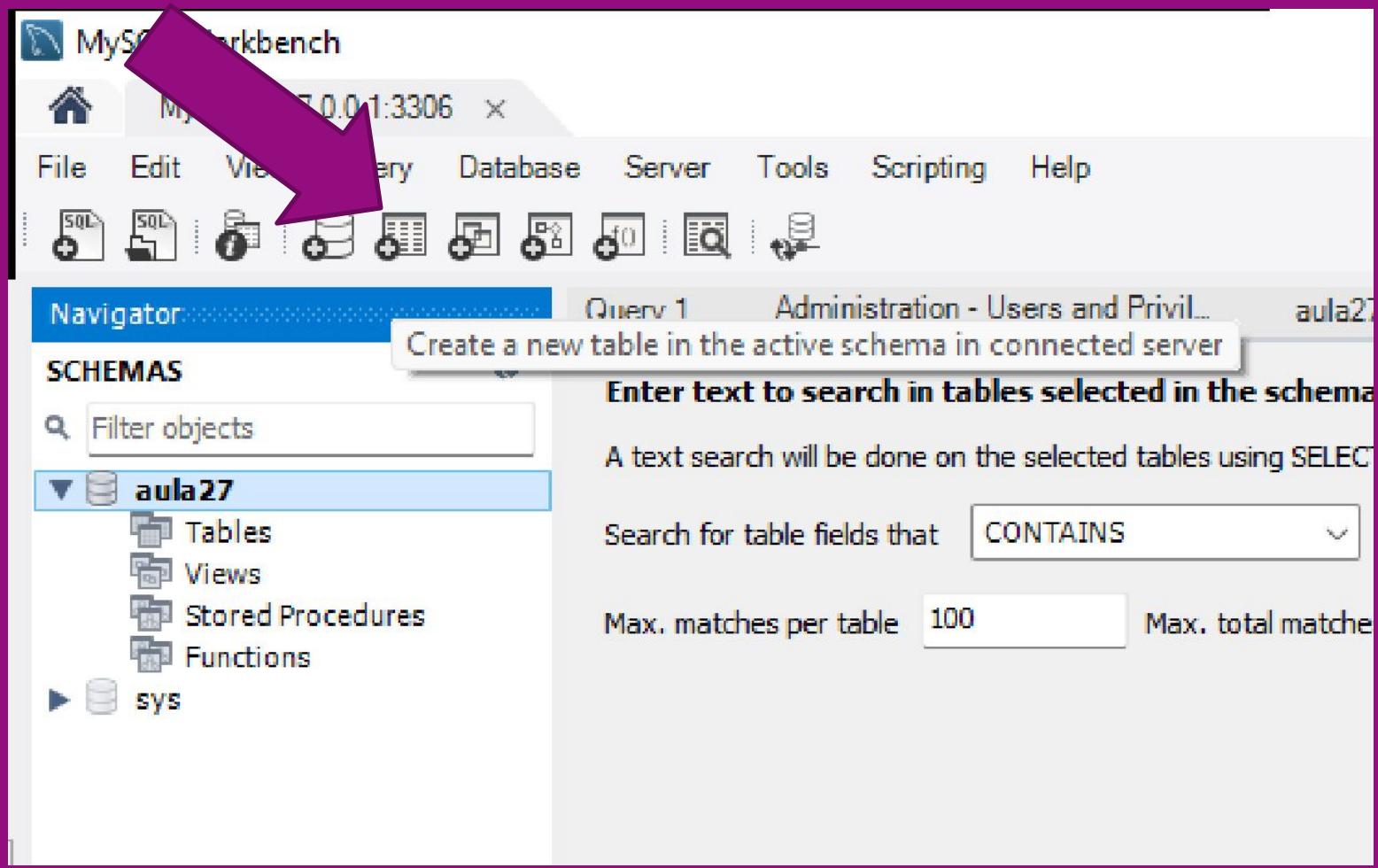


- Selecione a aba Schemas



- Botão direito no schema aula27 e acione a opção “Set as Default Schema”

Criar Tabela



Vamos usar nosso exemplo anterior

Campo

Unidade básica de uma tabela. Possui um nome, tipo e tamanho.

Exemplo: Nome, e-mail, endereço

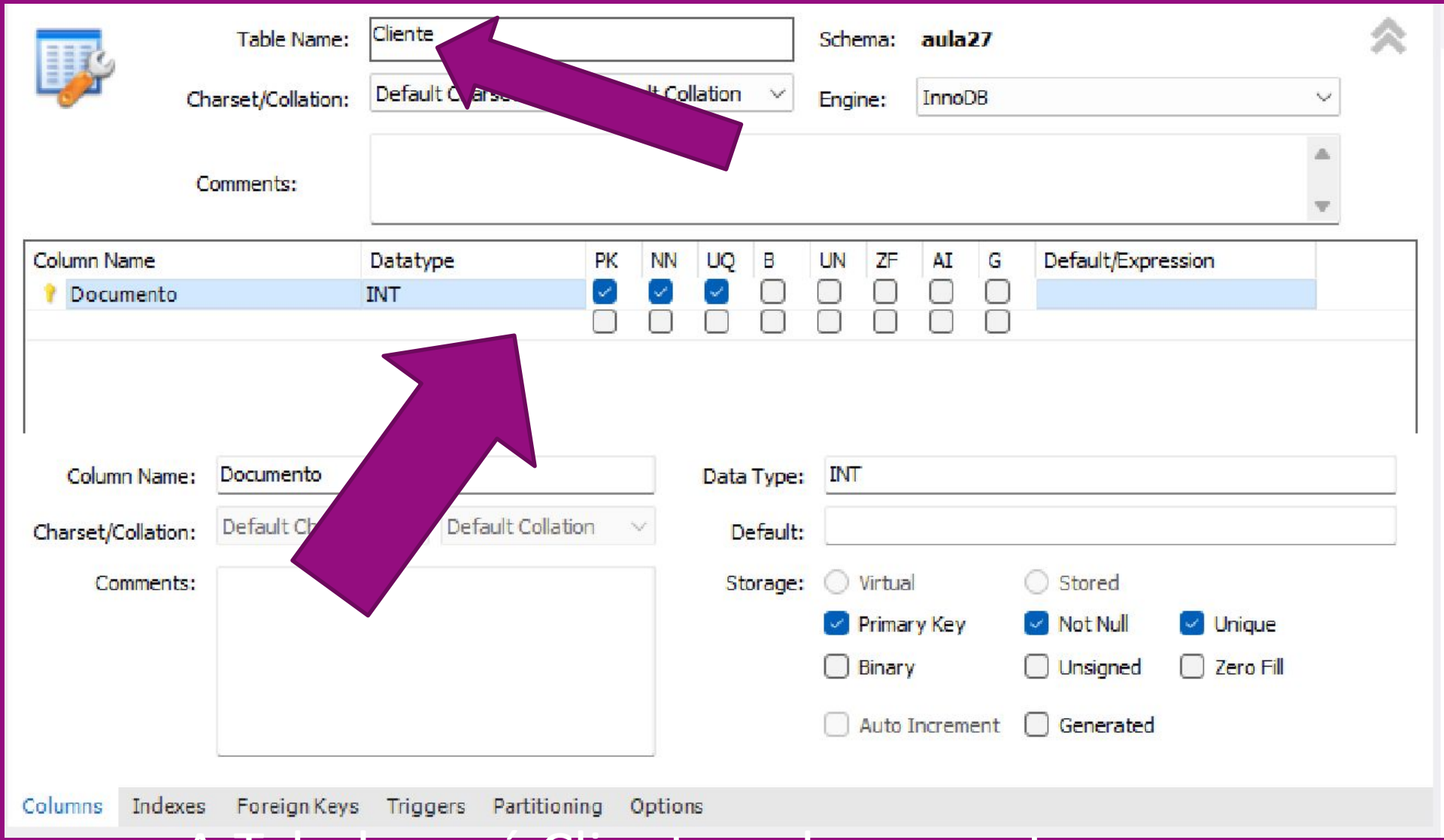
Tabela Lógica

União de linhas e colunas para armazenamento de dados dos sistemas.

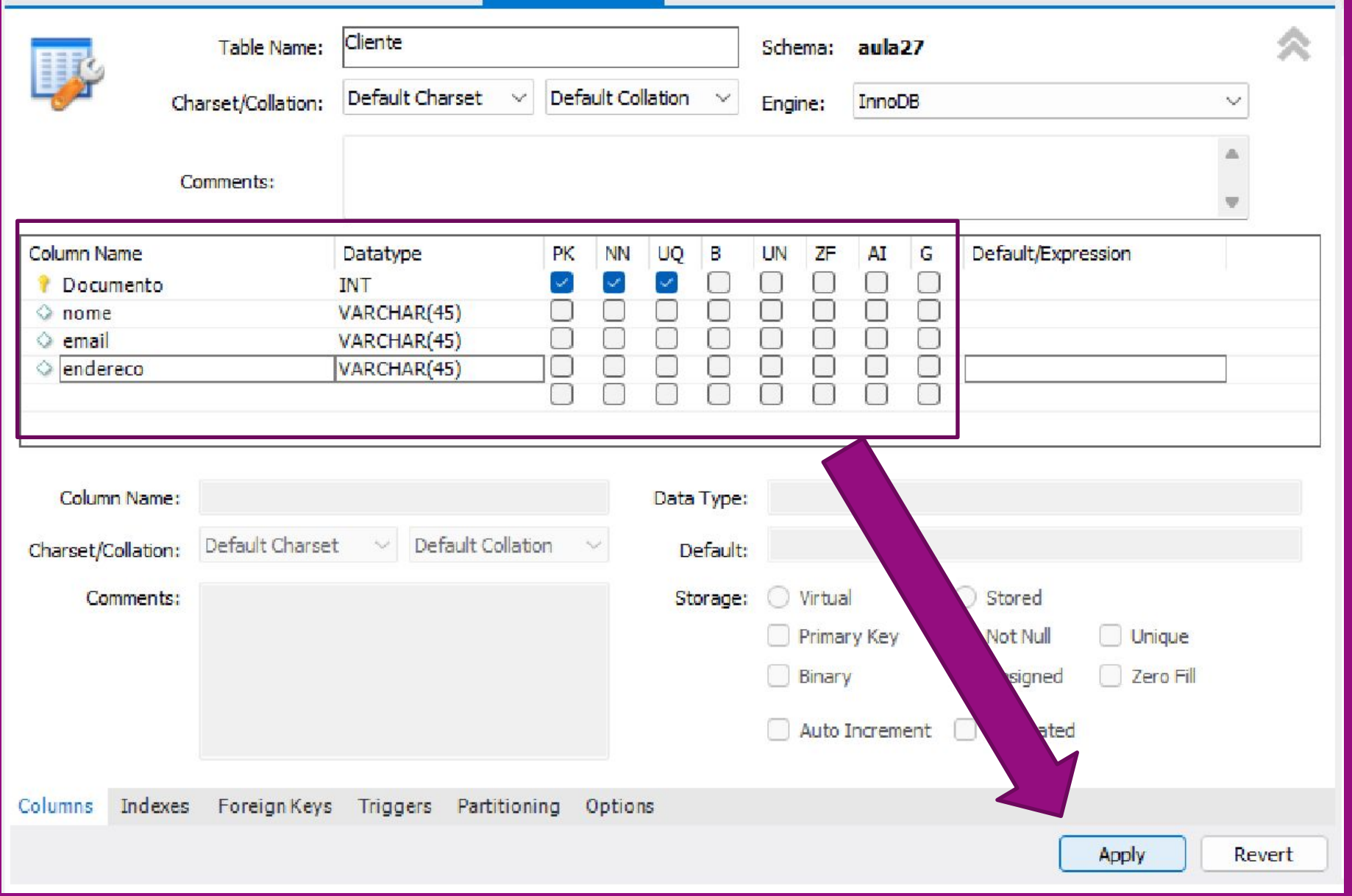
Exemplo: Tabela de clientes

Campos

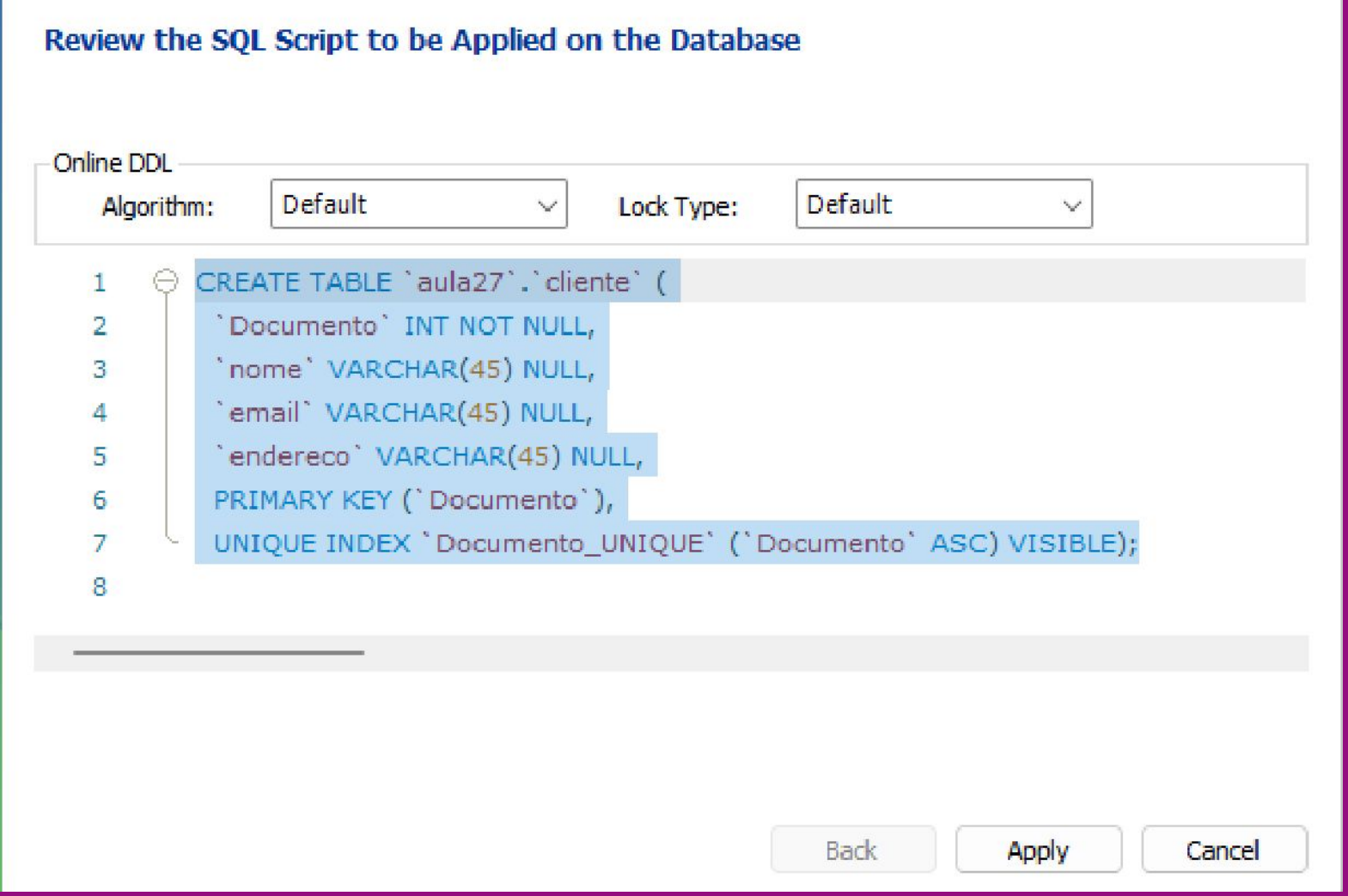
Documento	Nome	Email	Endereço
012.012-11	João da Silva	js@gmail.com	Rua Acacias
123.123-00	Maria da Silva	ma@gmail.com	Rua Acacias



- A Tabela será Cliente e documento a nossa chave primária



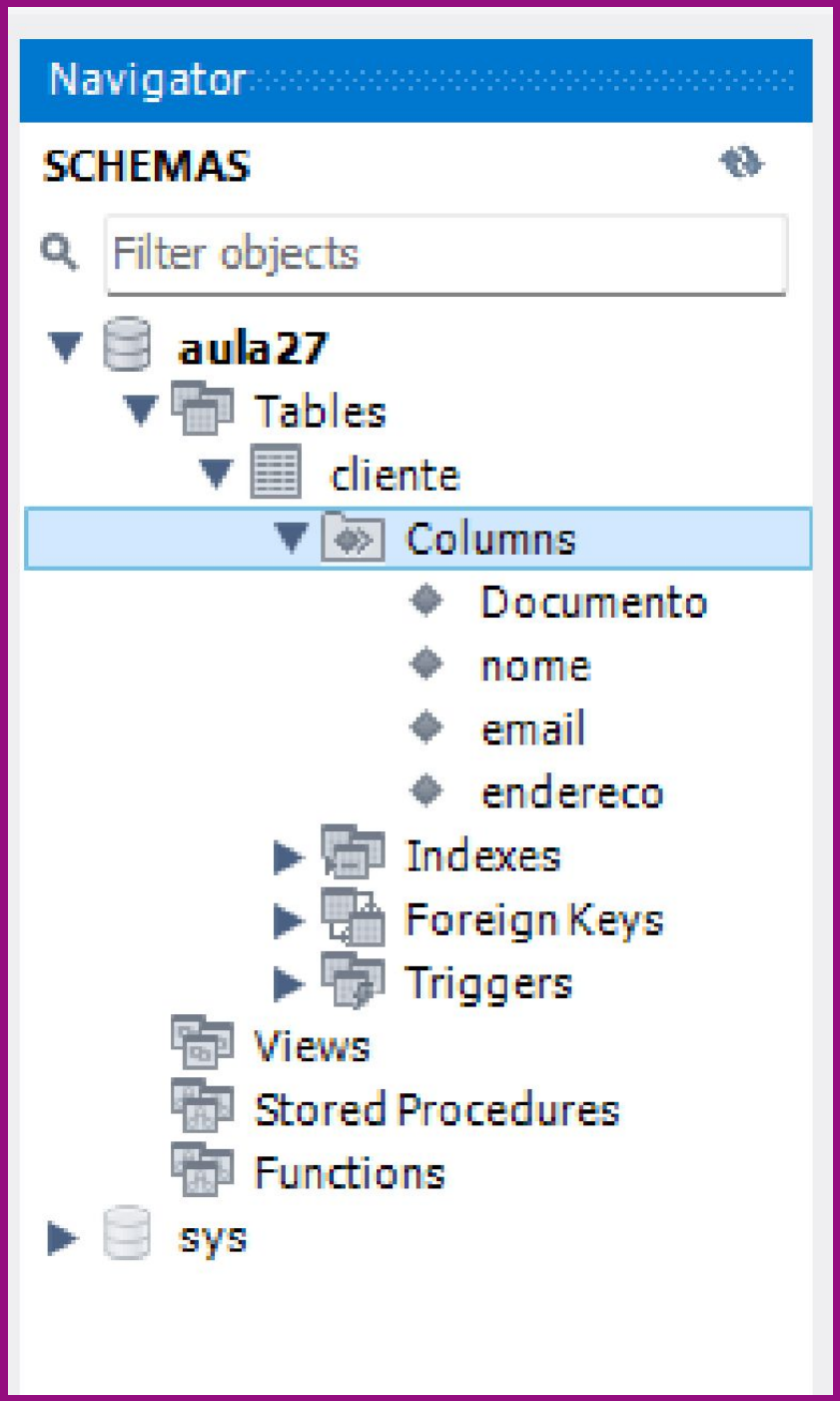
- Preencha os demais campos e clique em Apply



- Clique em Apply

A ferramenta gera o SQL

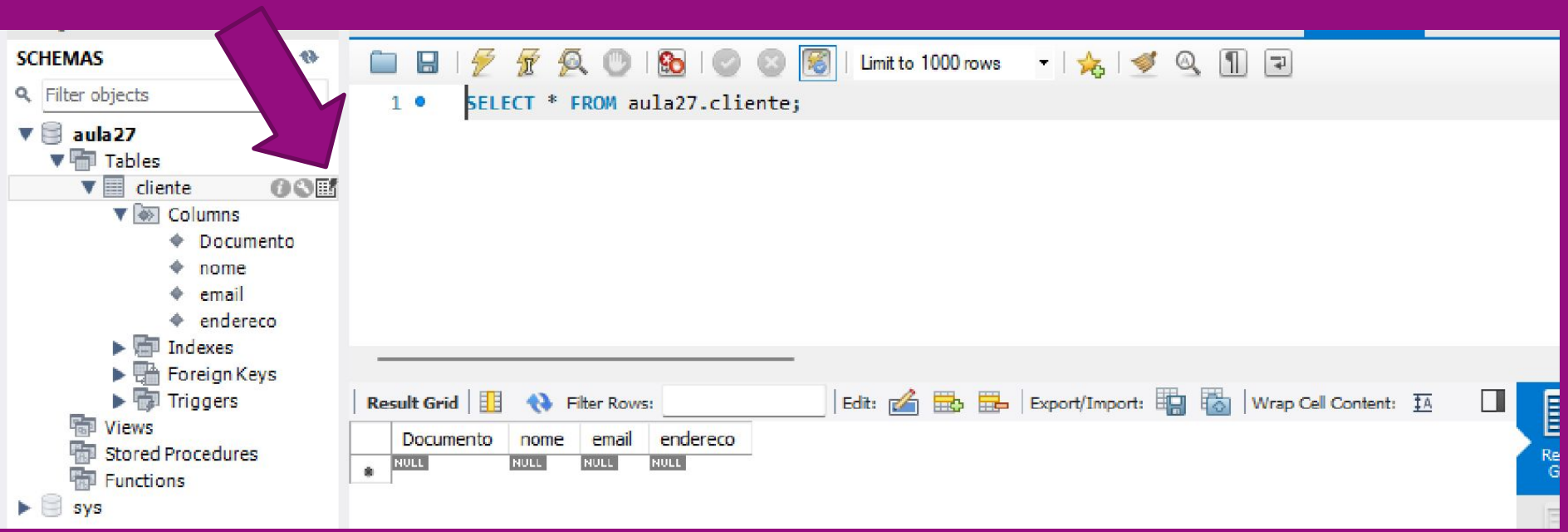
- CREATE TABLE `aula27`.`cliente` (`Documento` INT NOT NULL, `nome` VARCHAR(45) NULL, `email` VARCHAR(45) NULL, `endereco` VARCHAR(45) NULL, PRIMARY KEY (`Documento`), UNIQUE INDEX `Documento_UNIQUE` (`Documento` ASC) VISIBLE);



- Pronto
- Primeira Tabela criada

Select

- Clique no símbolo de tabela



Vamos usar nosso exemplo anterior

Campo

Unidade básica de uma tabela. Possui um nome, tipo e tamanho.

Exemplo: Nome, e-mail, endereço

Tabela Lógica

União de linhas e colunas para armazenamento de dados dos sistemas.

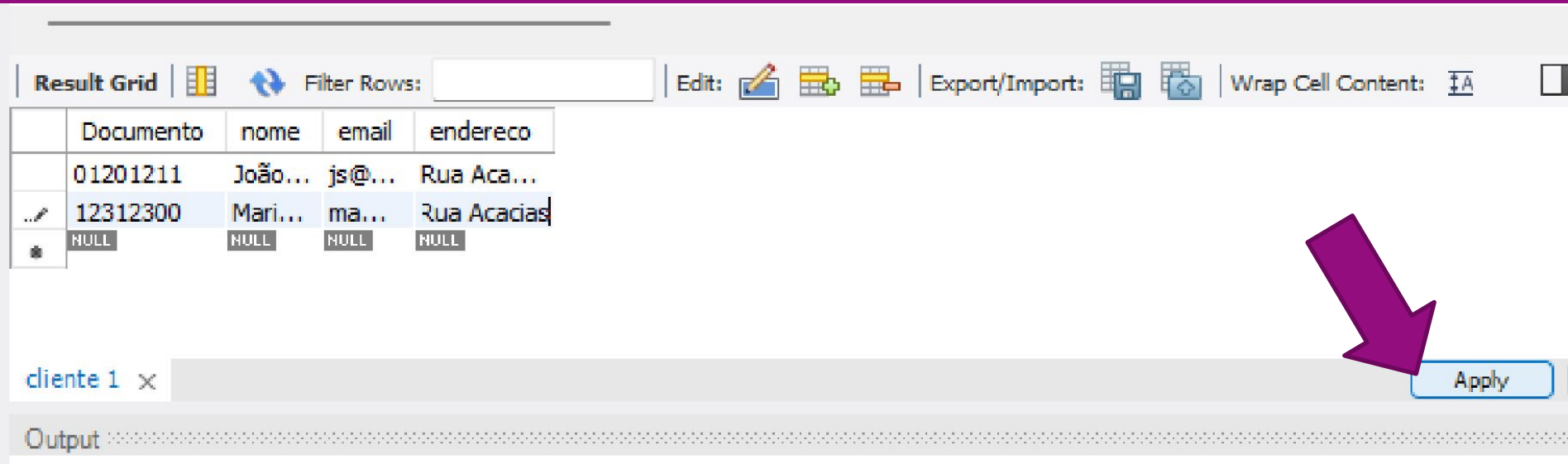
Exemplo: Tabela de clientes

Campos

Documento	Nome	Email	Endereço
012.012-11	João da Silva	js@gmail.com	Rua Acacias
123.123-00	Maria da Silva	ma@gmail.com	Rua Acacias

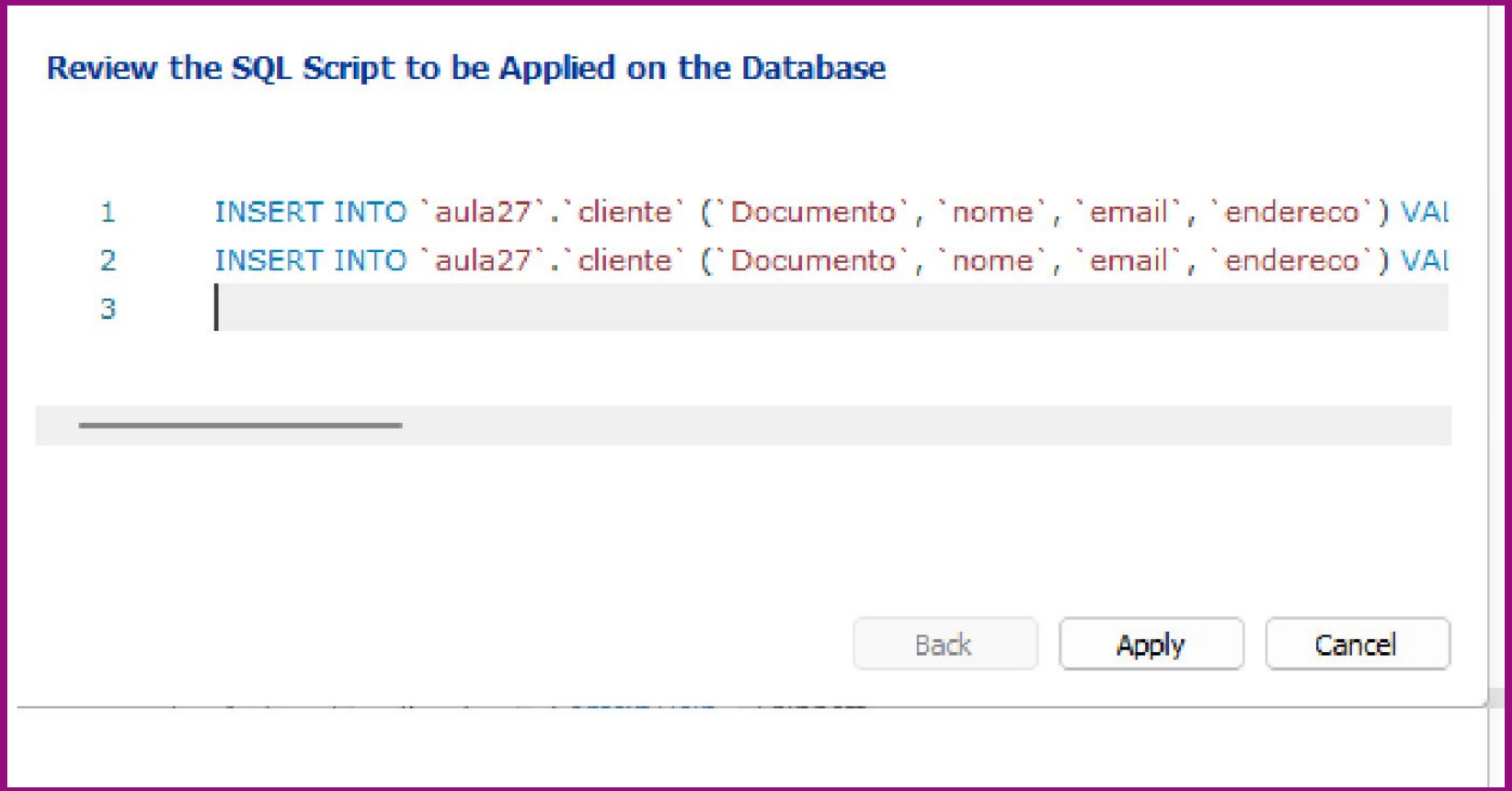
Salvando dados

- Preencha os dados conforme o exemplo e clique em Apply



Comando INSERT do SQL

- Confirme Apply

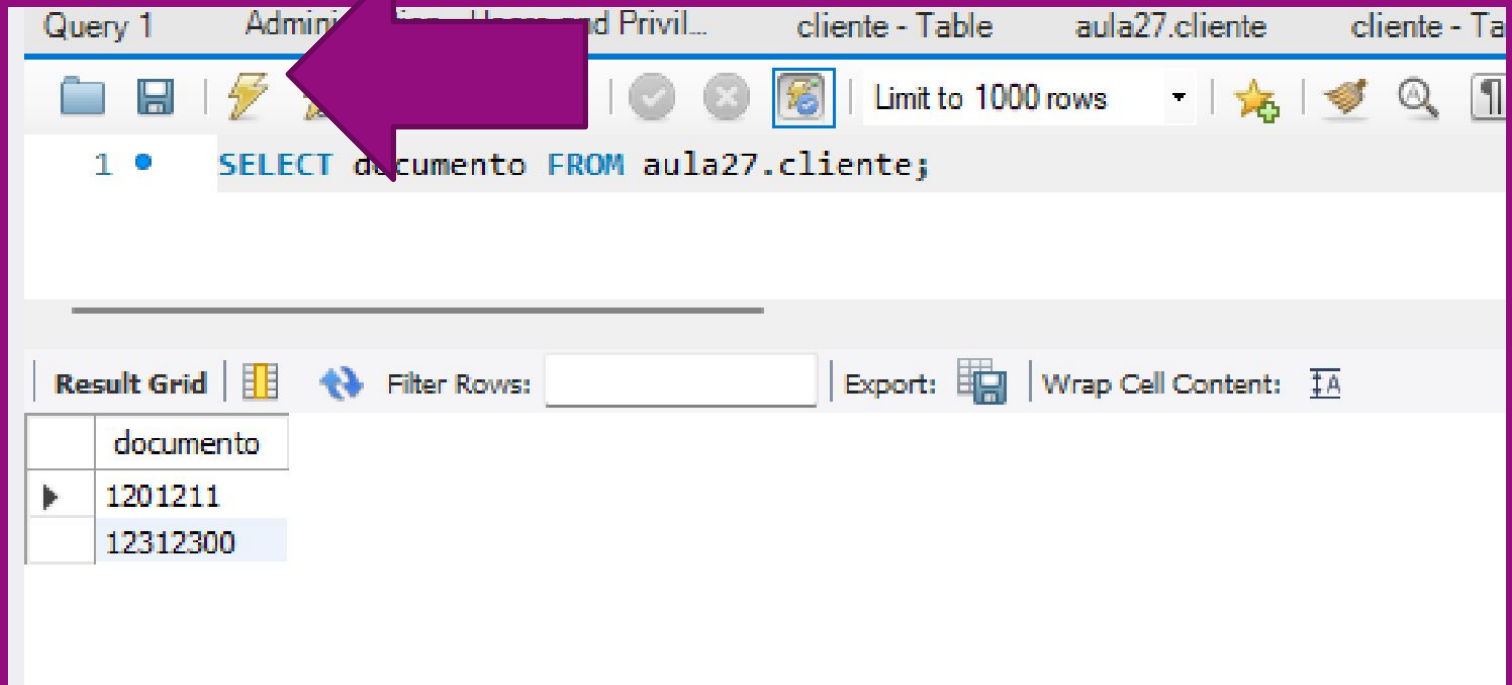


A Ferramenta gerou os “inserts” no banco

- INSERT INTO `aula27`.`cliente` (`Documento`, `nome`, `email`, `endereco`) VALUES ('01201211', 'João da Silva', 'js@gmail.com', 'Rua Acacias’);
- INSERT INTO `aula27`.`cliente` (`Documento`, `nome`, `email`, `endereco`) VALUES ('12312300', 'Maria da Silva', 'ma@gmail.com', 'Rua Acacias');

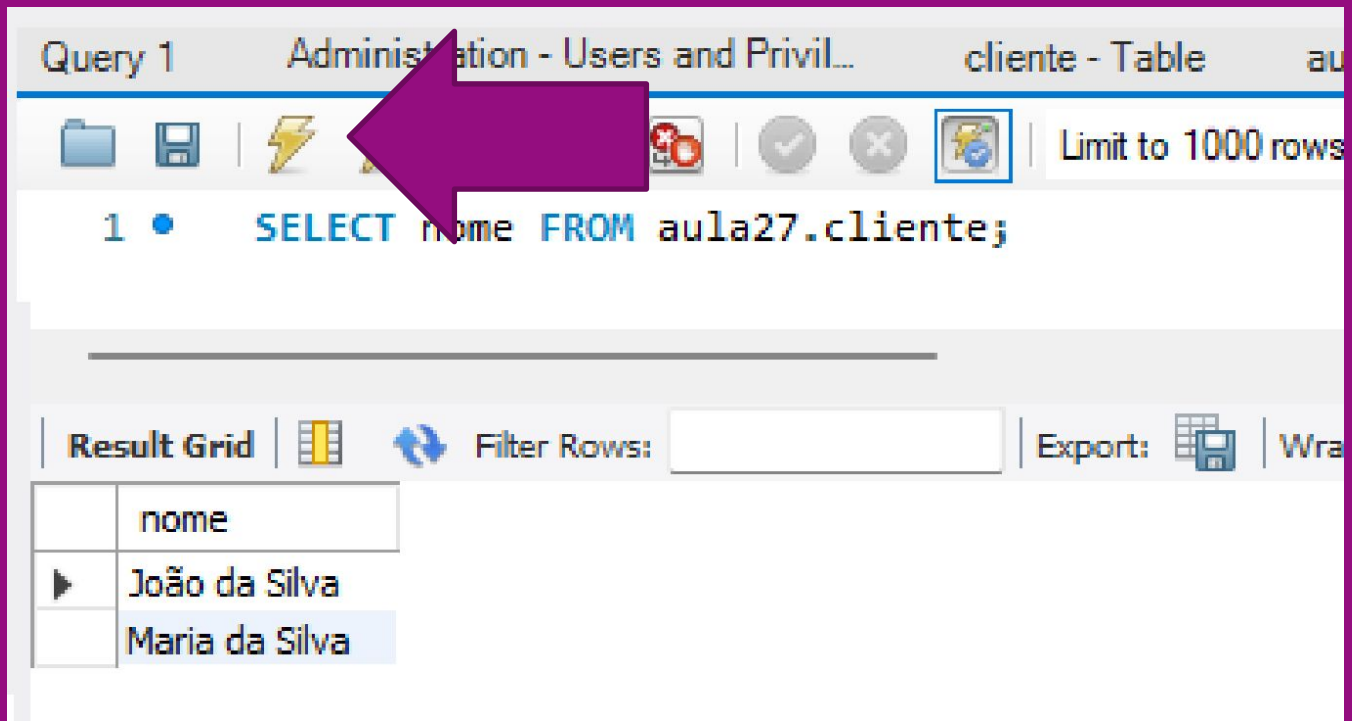
Agora podemos brincar

- SELECT Documento FROM aula27.cliente;
- Retorna apenas os Dados do campo Documento (Clique no símbolo do raio)



Agora podemos brincar

- SELECT nome FROM aula27.cliente;
- Retorna apenas os Dados do campo nome (Clique no símbolo do raio)

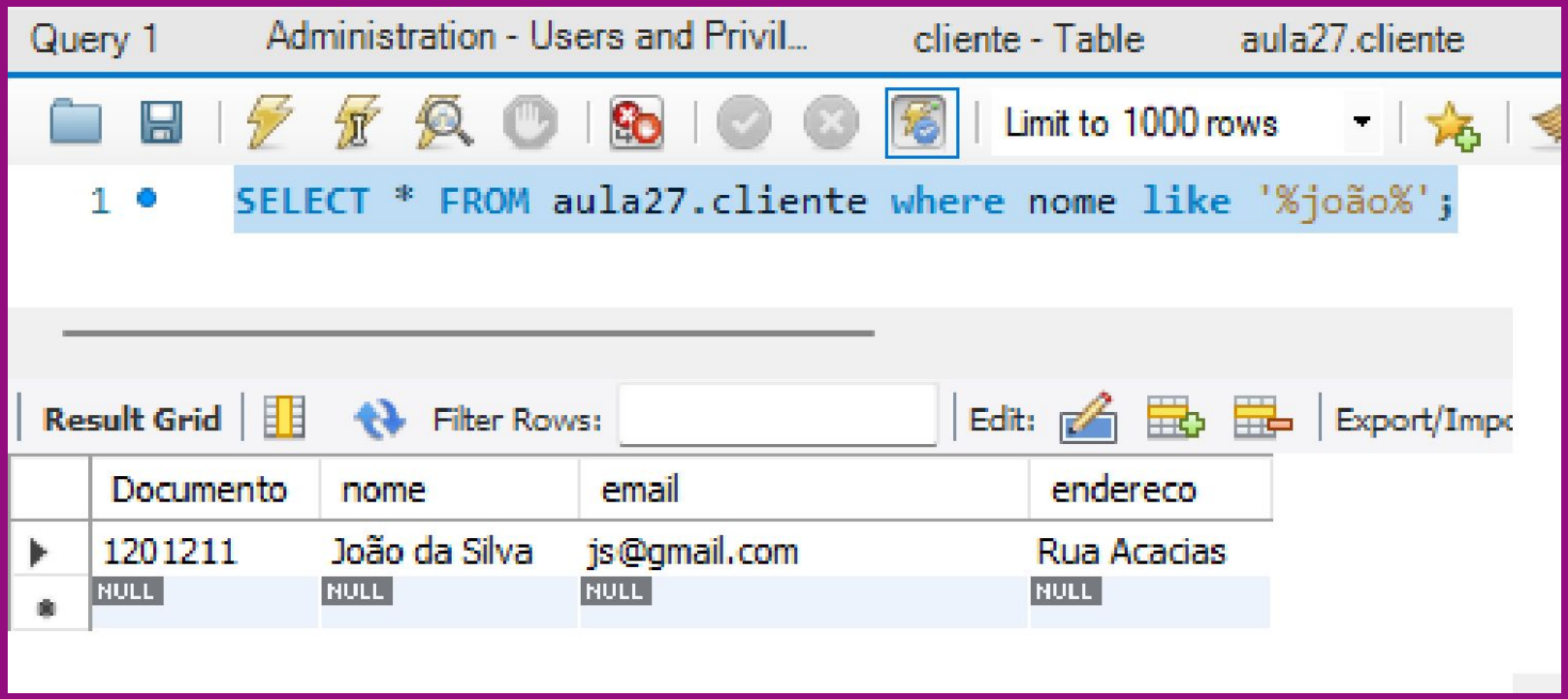


Exercício 1

- Faça o select para os demais campos
  - Endereco
  - email

Adiantando um pouco as coisas

- SELECT * FROM aula27.cliente where nome like '%joão%';



Adiantando um pouco as coisas

- SELECT * FROM aula27.cliente where nome like '%joão%’;
  - Where: Para especificar um parametro de busca
  - Quando for texto, deve estar entre ‘’
  - A palavra “like” neste caso serve para dizer para retornar qualquer registro que contenha a string. Sem o Like teria que passar o nome exatamente como está no registro, por exemplo ‘João da Silva’

Atividade 1: Crie sua tabela para representar este formulário

Carga horária	**Semestre**	**Ano**
160h	2º	2020
Nome dos professores:		
Cursos:		
Bacharelado em Ciência da Computação, Bacharelado em Sistemas de Informação, Análise e Desenvolvimento de Sistemas, Banco de Dados, Jogos Digitais, Sistemas para Internet e Gestão de Tecnologia da Informação		

Atividade 2: Crie sua tabela para representar este formulário

Nome Aluno
Prova 1
Prova 2
Prova 3
Freq. (%)
Aproveit. Semestral
Aval. Final
ResultFinal

Vamos Conferir

?

Gabarito

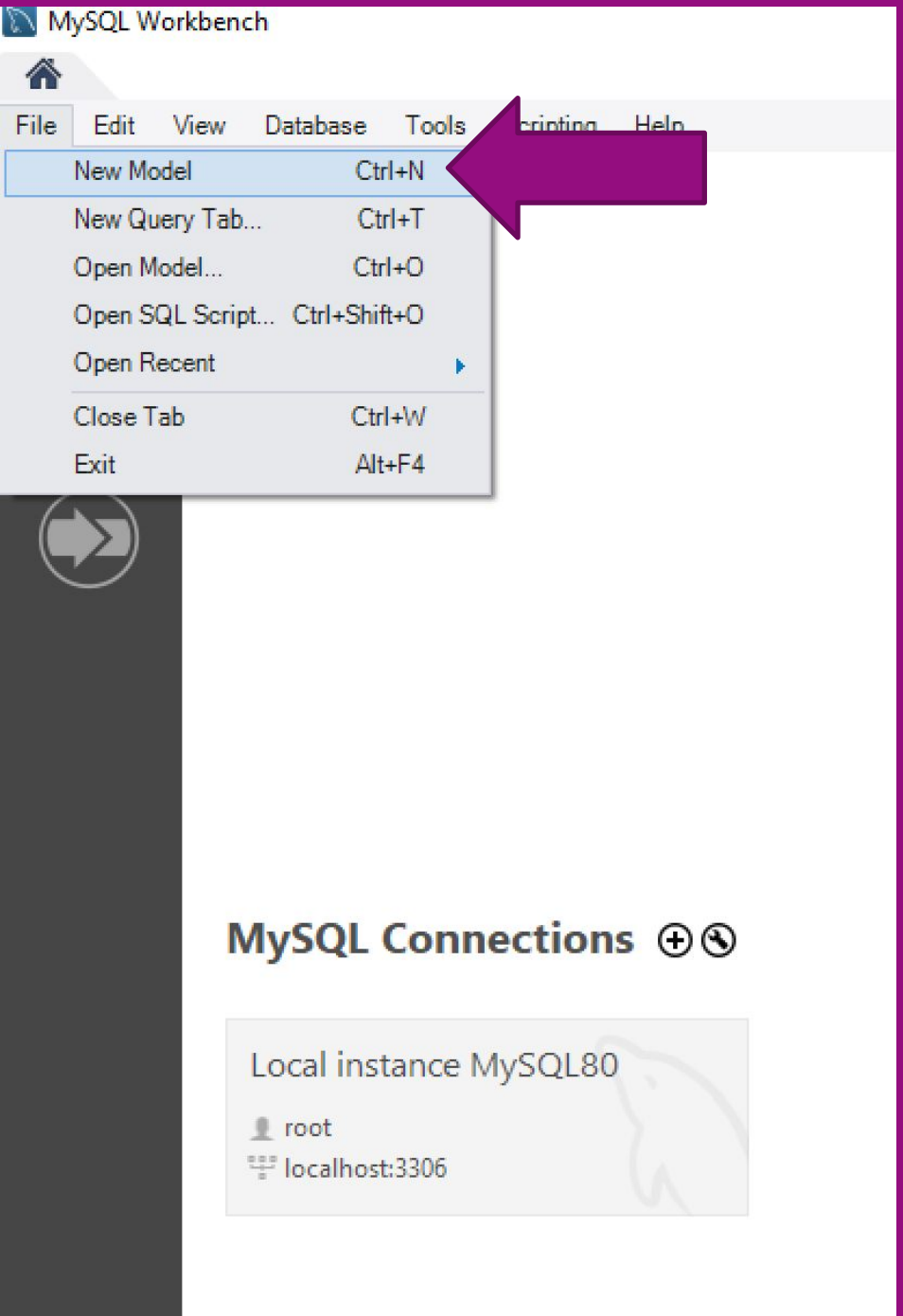
- CREATE TABLE `aula27`.`professores` (`idProfessores` INT NOT NULL, `cargahorarica` INT NULL, `semestre` INT NULL, `ano` INT NULL, `nome_professores` LONGTEXT NULL, `cursos` LONGTEXT NULL, PRIMARY KEY (`idProfessores`), UNIQUE INDEX `idProfessores_UNIQUE` (`idProfessores` ASC) VISIBLE);

Gabarito

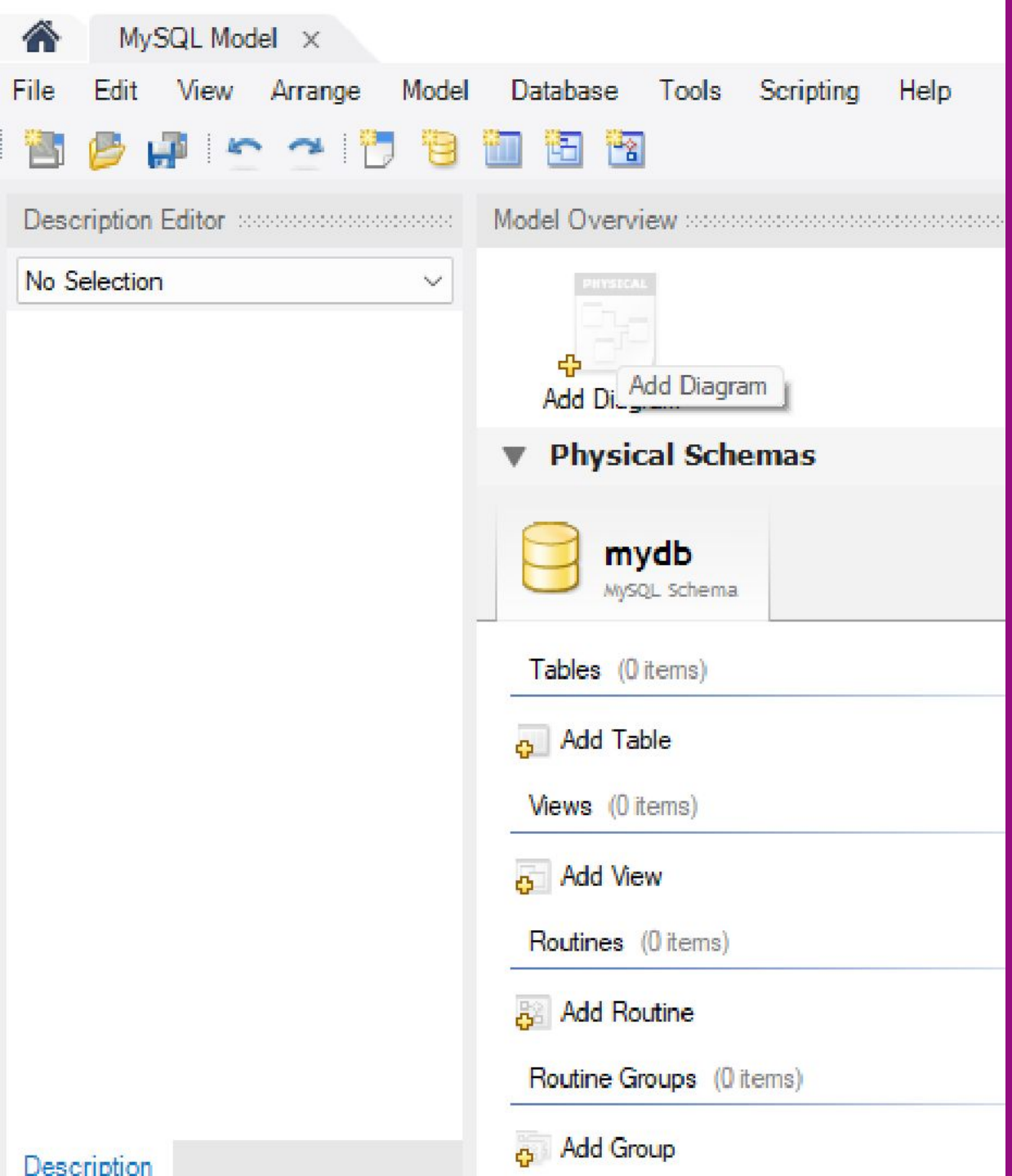
- CREATE TABLE `aula27`.`aluno` (`matricula` INT NOT NULL, `prova1` FLOAT NULL, `prova2` FLOAT NULL, `prova3` FLOAT NULL, `frequencia` FLOAT NULL, `aproveitamento_semestral` FLOAT NULL, `avaliacao_final` FLOAT NULL, `resultado_final` VARCHAR(45) NULL, PRIMARY KEY (`matricula`));

Busca Ativa

- Leitura do Livro:
  - Barboza, Fabrício Felipe, M. e Pedro Henrique Chagas Freitas. *Modelagem e desenvolvimento de banco de dados*. Disponível em: Minha Biblioteca, Grupo A, 2018



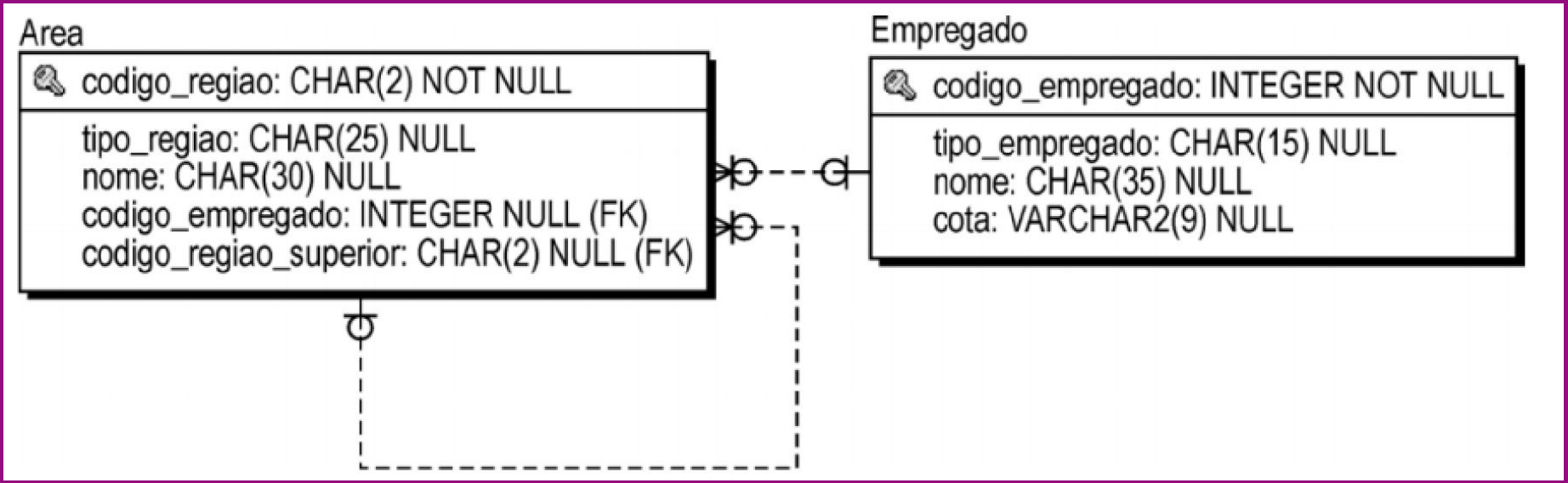
Agora Vamos Modelar



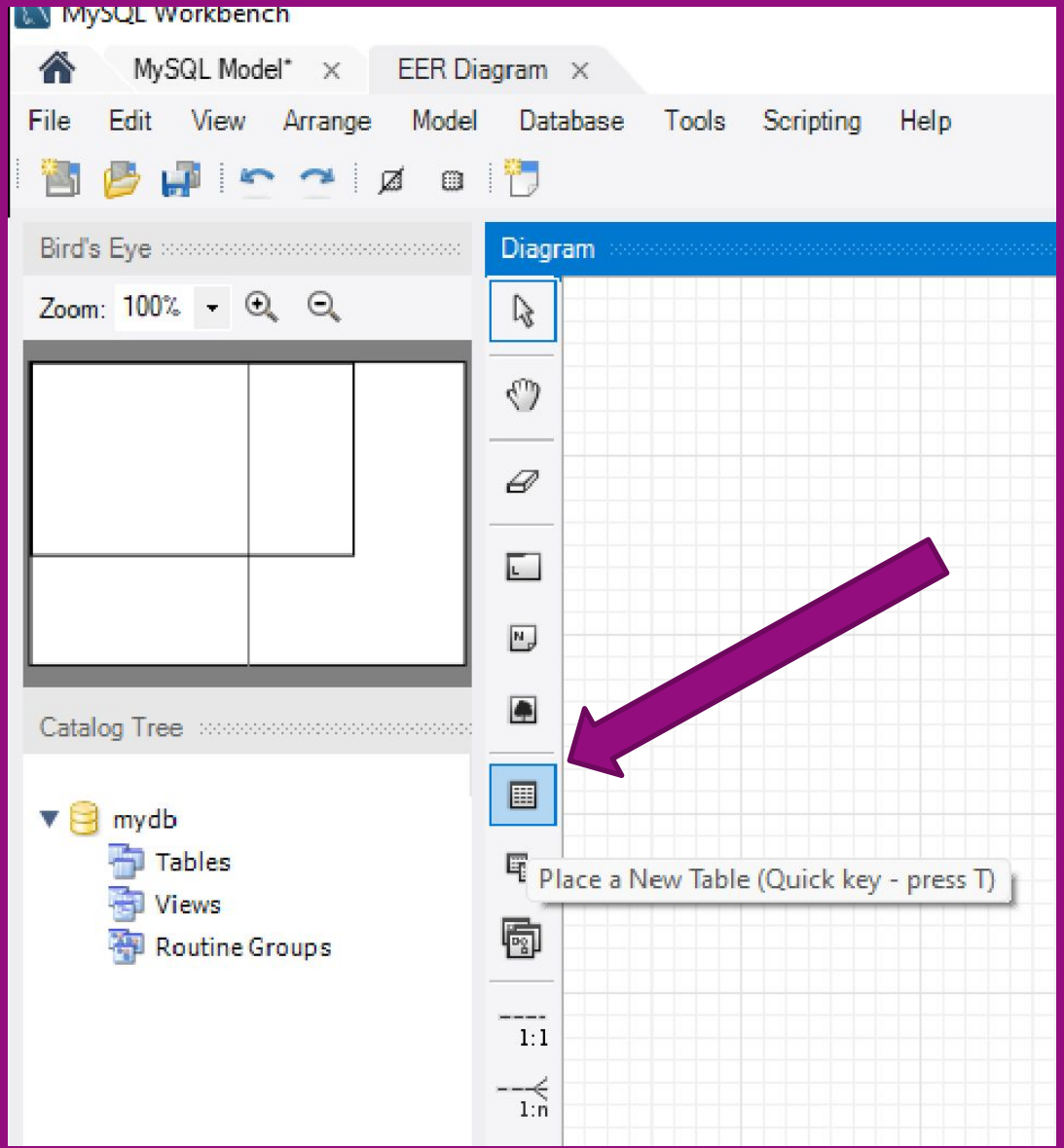
- Adicione um diagrama

Exemplo
Modelo físico

- Página 22 do livro:
  - Machado, Felipe Nery R. *BANCO DE DADOS – PROJETO E IMPLEMENTAÇÃO*. Disponível em: Minha Biblioteca, (4th edição). Editora Saraiva, 2020.



Vamos Criar o nosso modelo



- Crie uma Tabela

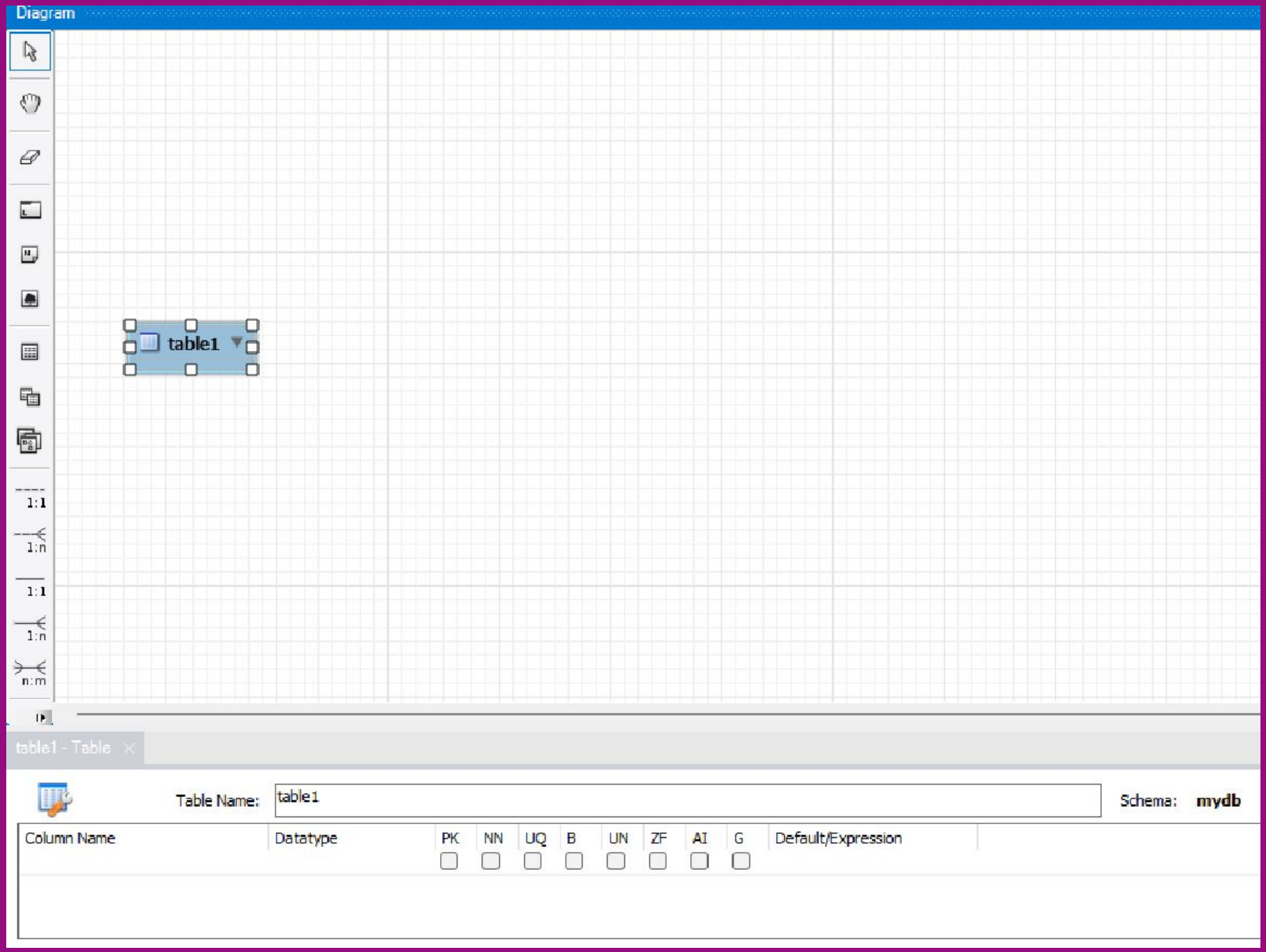
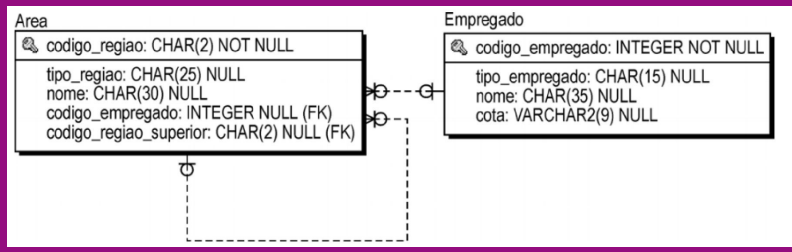


Diagram
table1
1:1
1:n
1:1
1:n
n:m
table1 - Table
Table Name:
table1
Schema:
mydb
Column Name
Datatype
PK
NN
UQ
B
UN
ZF
AI
G
Default/Expression

Altere o nome da tabela conforme o exemplo



MySQL Workbench

MySQL Model* | EER Diagram

File Edit View Arrange Model Database Tools Scripting Help

Bird's Eye | Diagram

Zoom: 100%

Catalog Tree

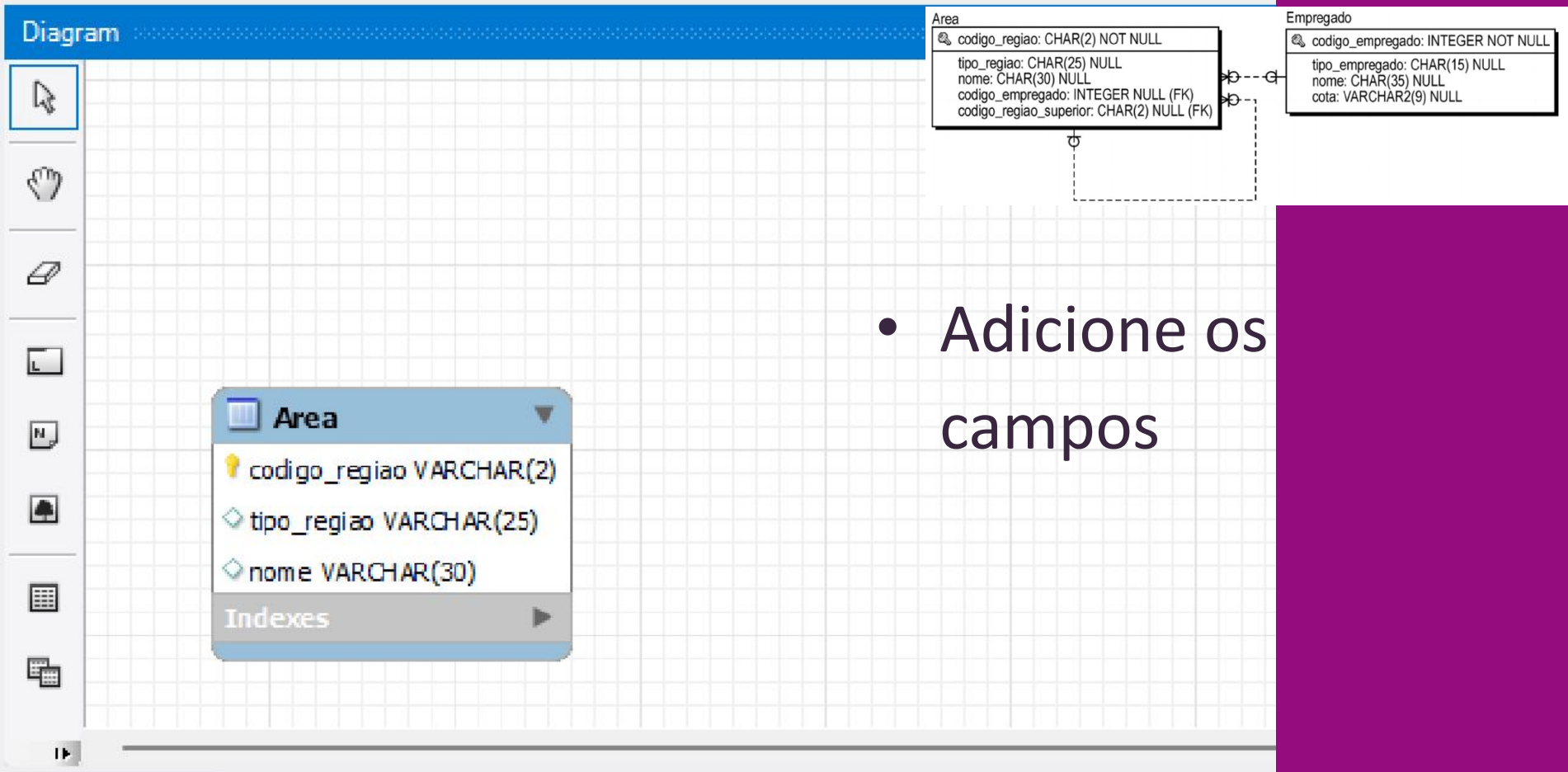
- mydb
  - Tables
  - Views
  - Routine Groups

Area

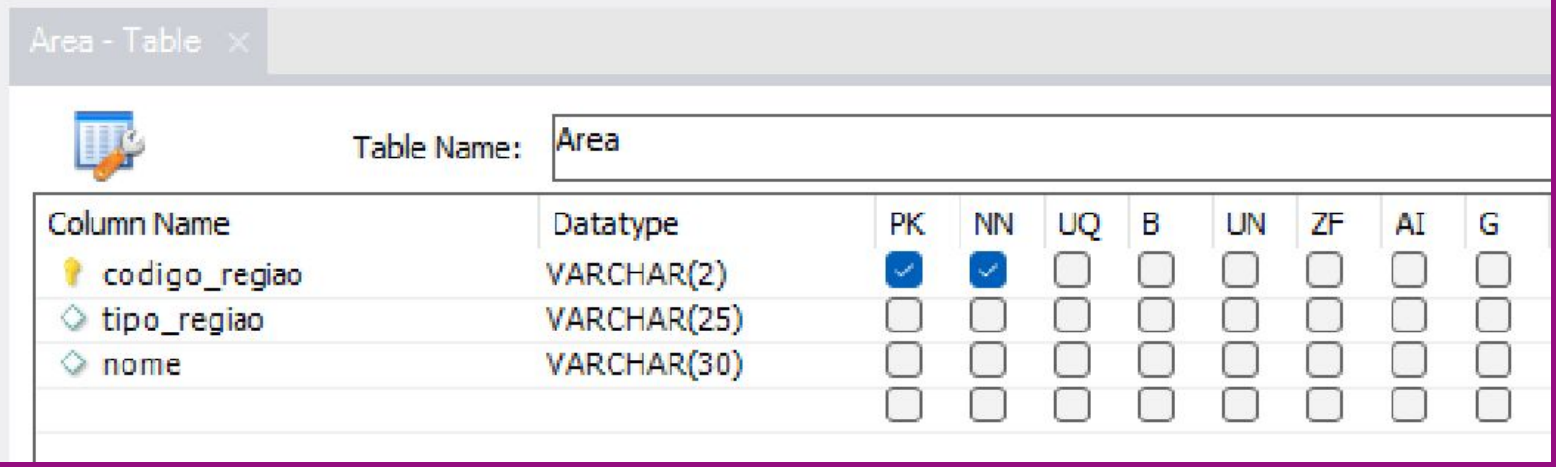
Area - Table

Table Name: Area

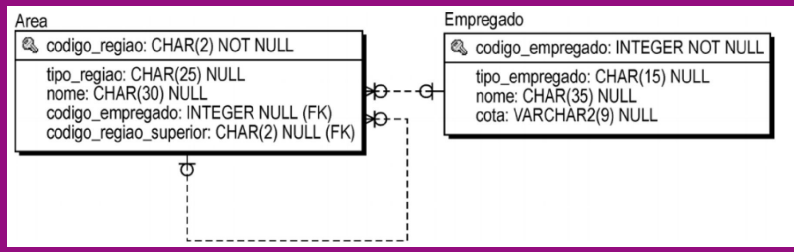
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression

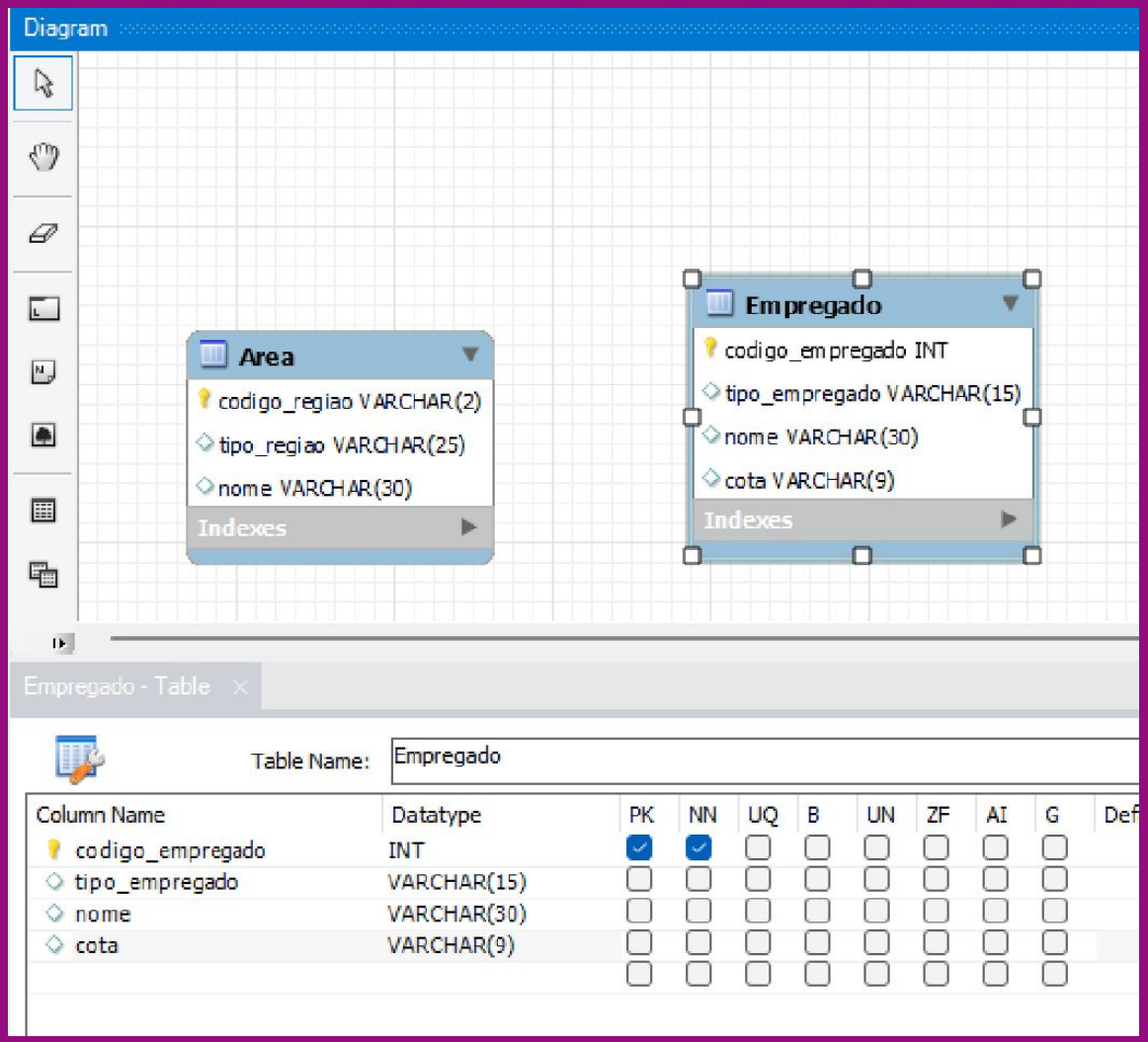


- Adicione os campos

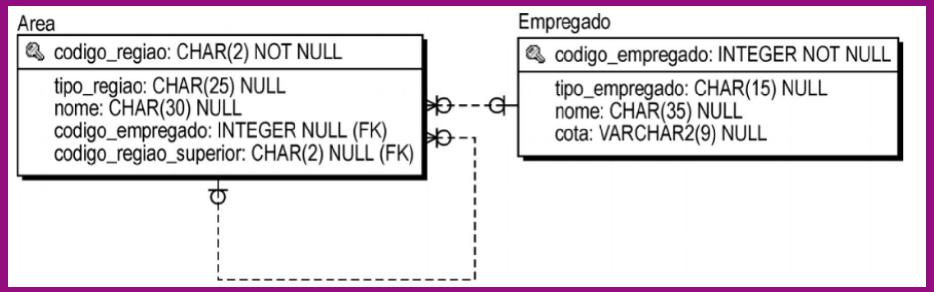


Crie a segunda Tabela

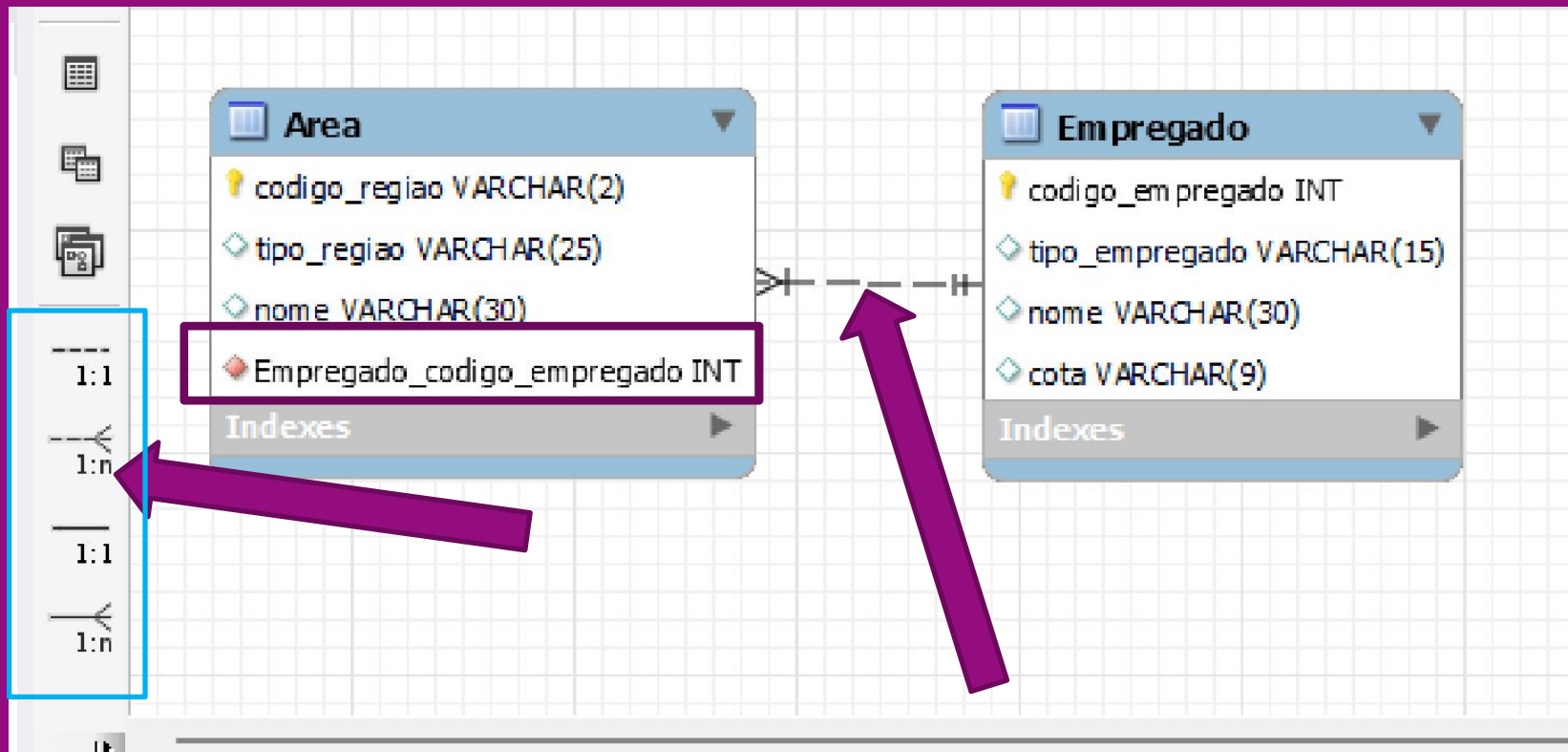




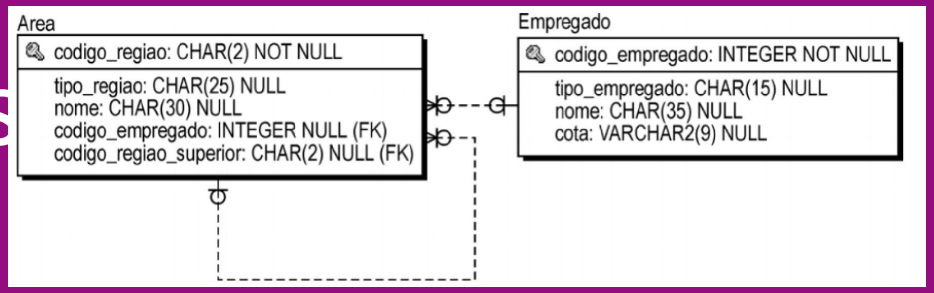
Relacionamentos



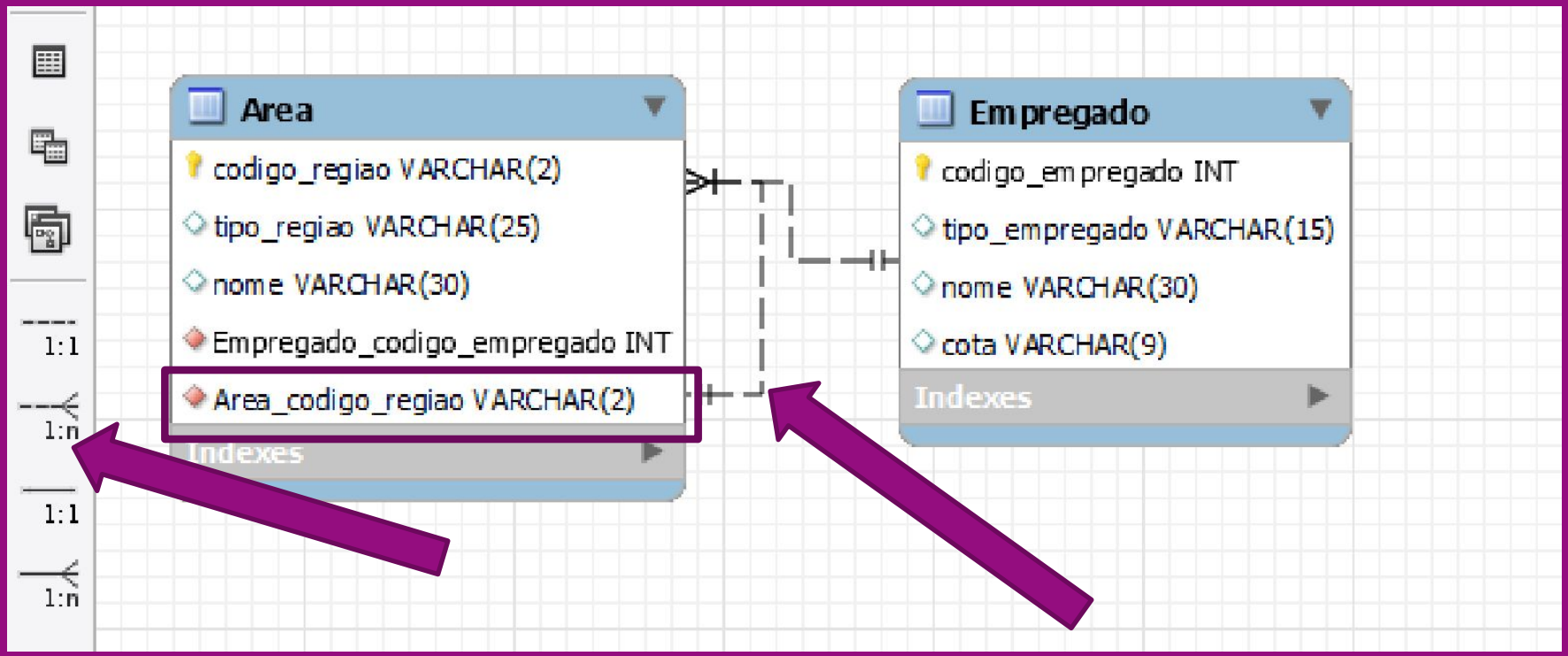
- Note que ao criar o relacionamento, o campo da chave estrangeira é criado automaticamente.



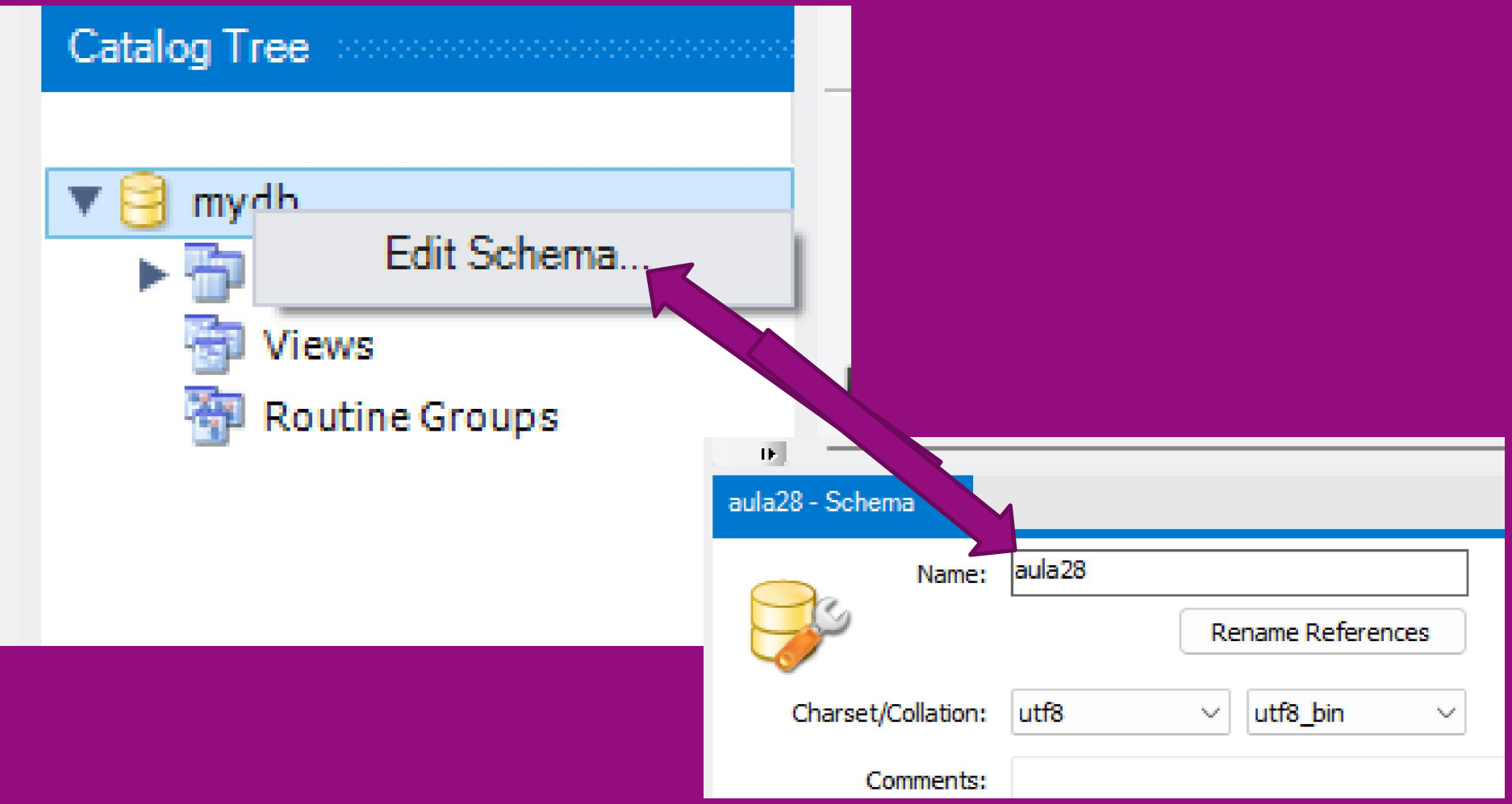
Relacionamentos



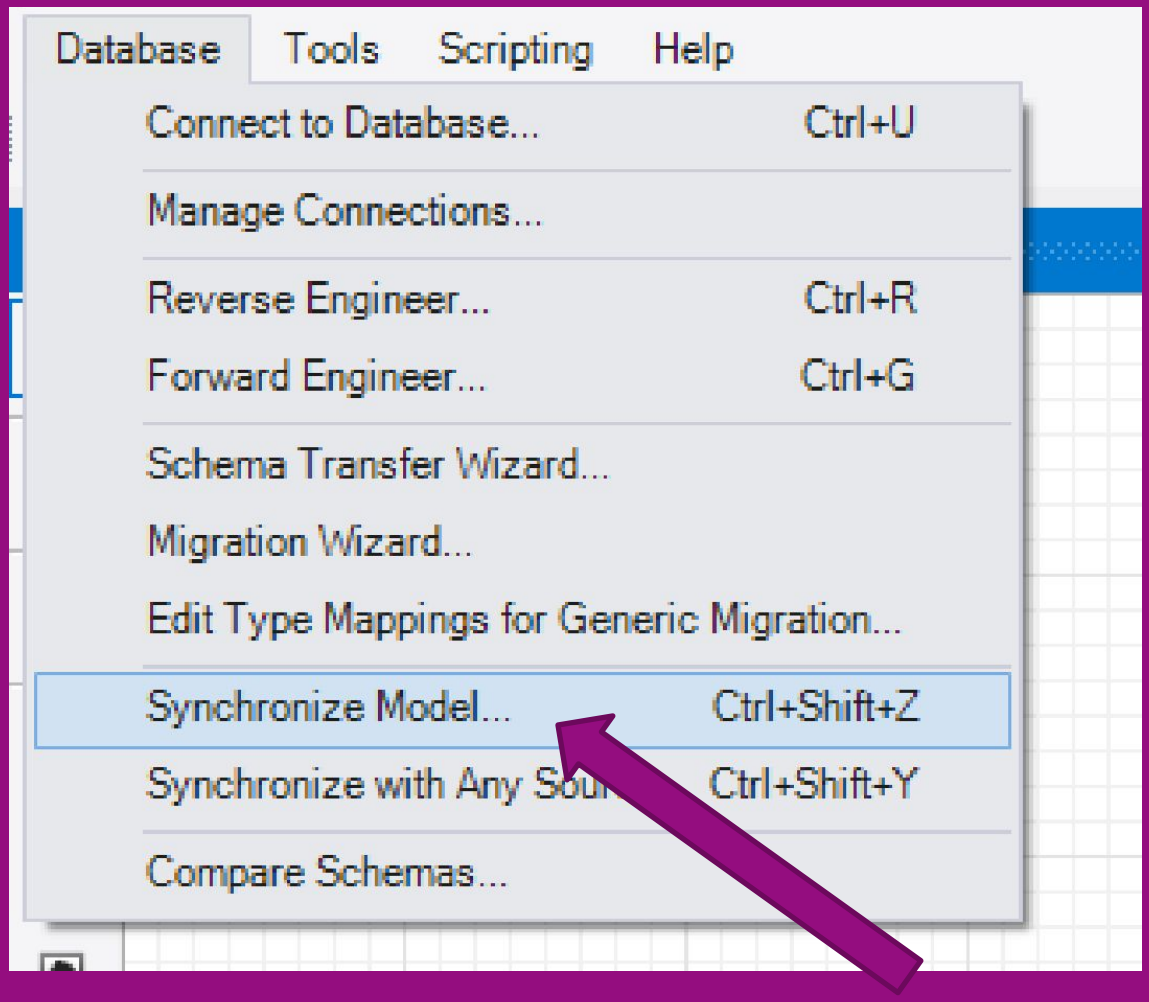
- Crie o segundo relacionamento



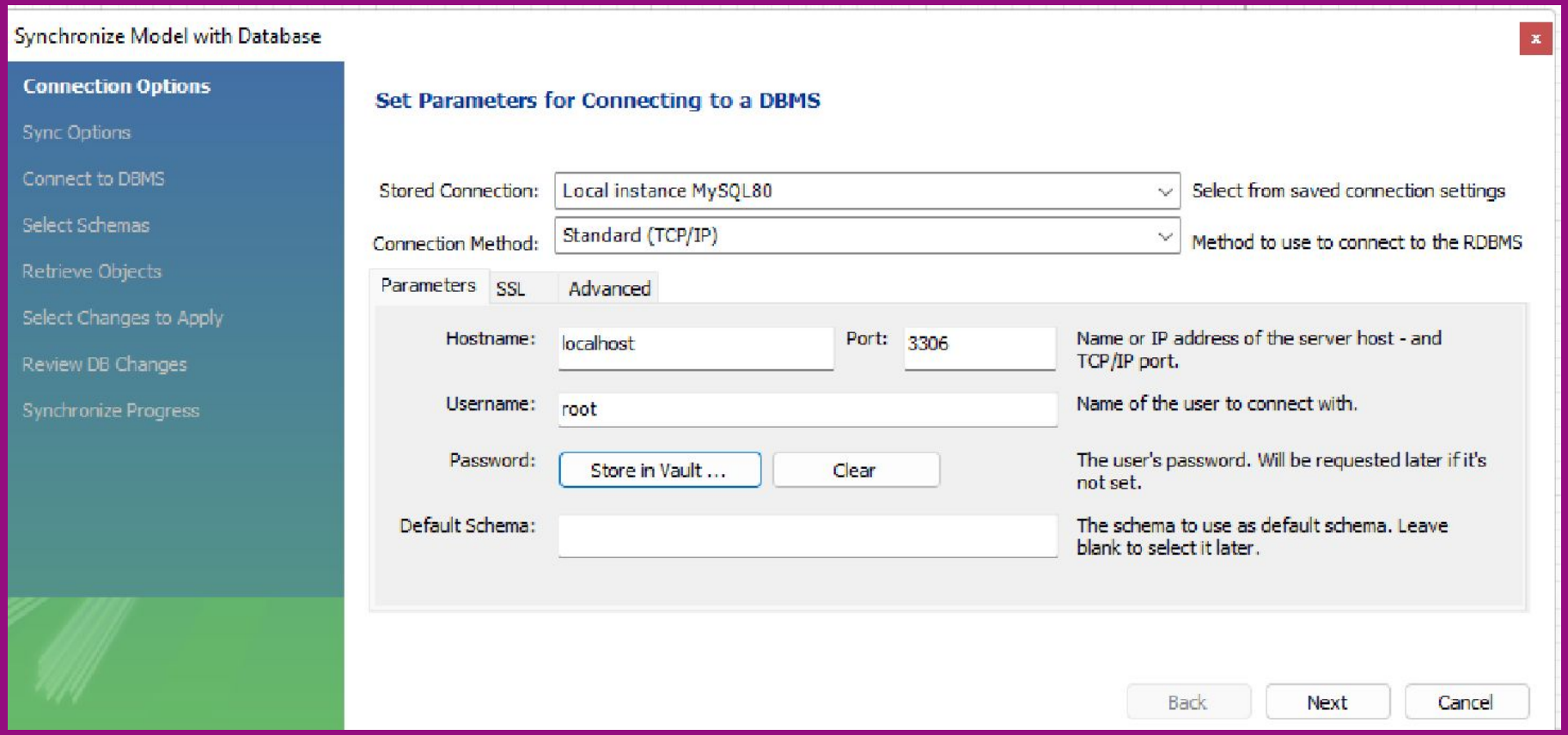
Vamos criar um Schema



Vamos criar as tabelas no nosso Schema



Conectar com o *database*



Selecionar o schema

Synchronize Model with Database

- Connection Options
- Sync Options
- Connect to DBMS
- Select Schemas
- Retrieve Objects
- Select Changes to Apply
- **Review DB Changes**
- Synchronize Progress

Preview Database Changes to be Applied

```
1   -- MySQL Workbench Synchronization
2   -- Generated: 2022-05-27 09:49
3   -- Model: New Model
4   -- Version: 1.0
5   -- Project: Name of the project
6   -- Author: RIHS
7
8   SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
9   SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
10  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,N
11
12  CREATE SCHEMA IF NOT EXISTS `aula28` DEFAULT CHARACTER SET utf8 ;
13
14  CREATE TABLE IF NOT EXISTS `aula28`.`Area` (
15    `codigo_regiao` VARCHAR(2) NOT NULL,
16    `tipo_regiao` VARCHAR(25) NULL DEFAULT NULL,
17    `nome` VARCHAR(30) NULL DEFAULT NULL,
18    `Empregado_codigo_empregado` INT(11) NOT NULL,
19    `Area_codigo_regiao` VARCHAR(2) NOT NULL,
20    PRIMARY KEY (`codigo_regiao`),
21    INDEX `fk_Area_Empregado_idx` (`Empregado_codigo_empregado` ASC) VISIBLE,
22    INDEX `fk_Area_Area1_idx` (`Area_codigo_regiao` ASC) VISIBLE,
23    CONSTRAINT `fk_Area_Empregado`
24      FOREIGN KEY (`Empregado_codigo_empregado`)
25      REFERENCES `aula28`.`Empregado` (`codigo_empregado`)
26      ON DELETE NO ACTION
```

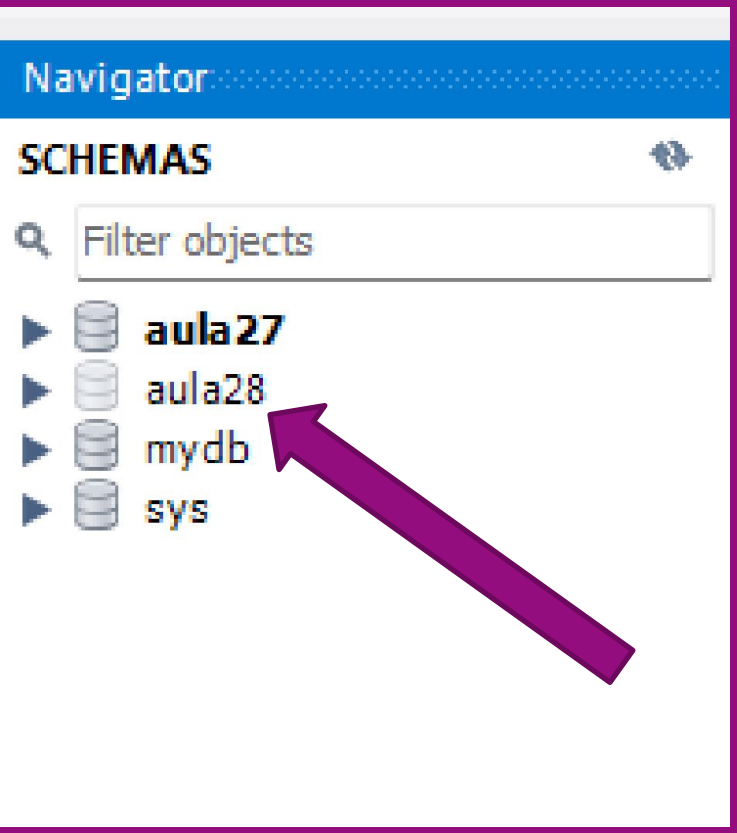
Save to File... | Copy to Clipboard | ☐ Skip DB changes and update model only

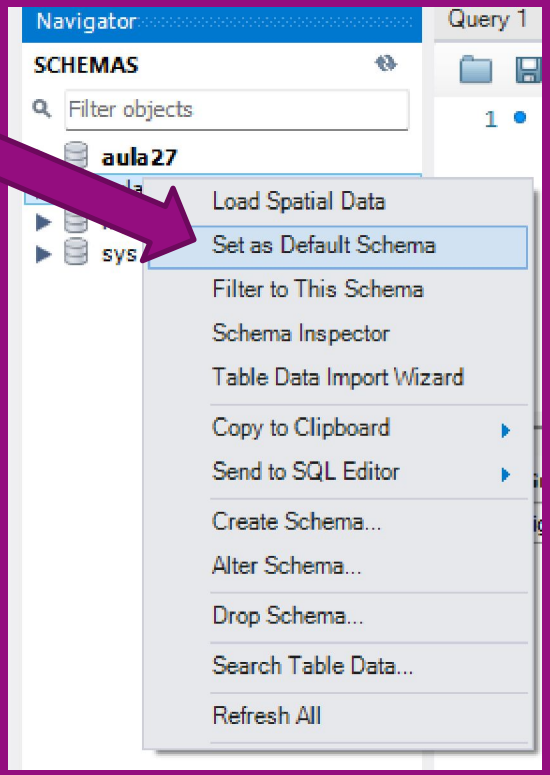
Back | Execute > | Cancel

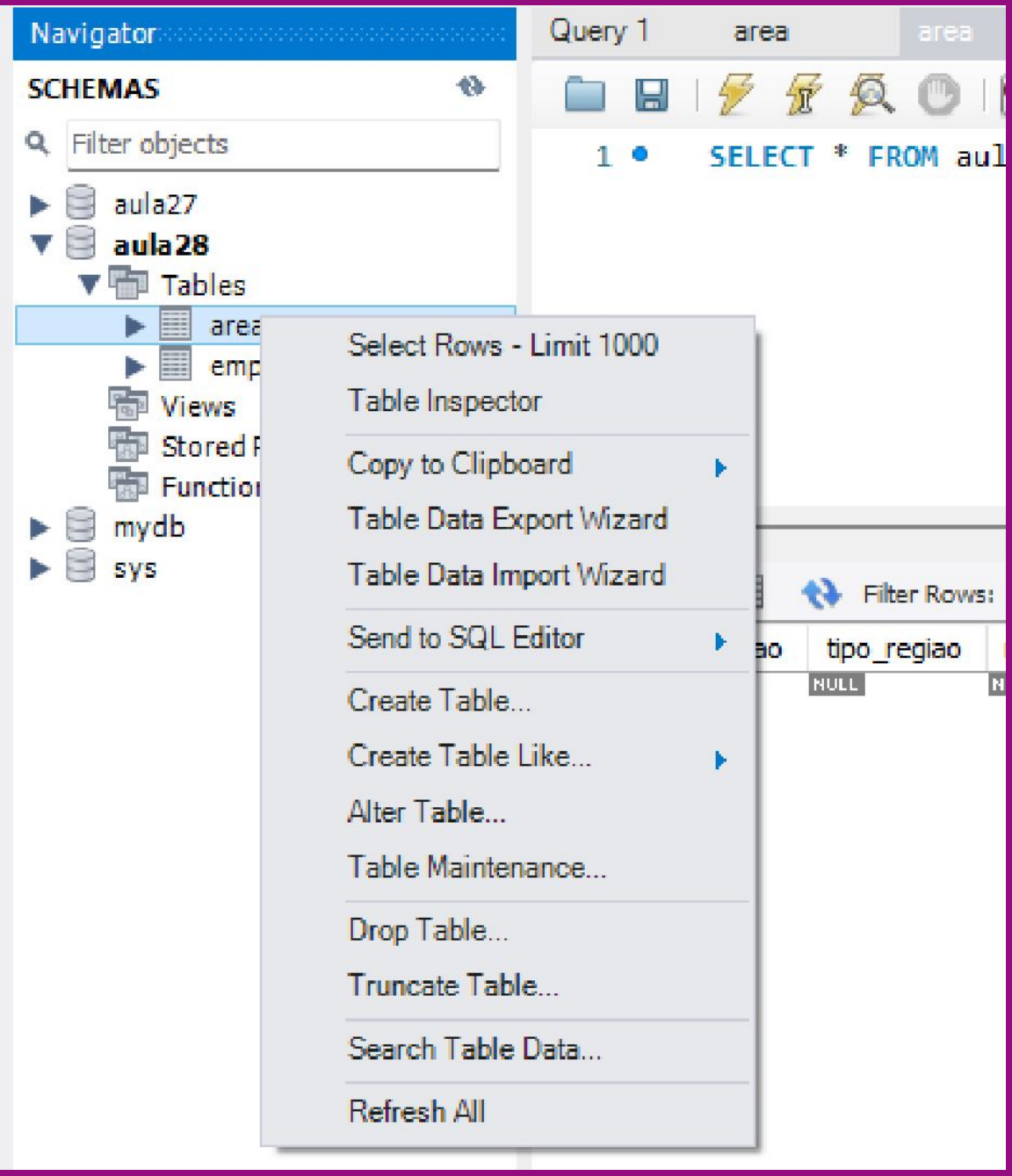
Create Table a partir do modelo

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';CREATE SCHEMA IF NOT EXISTS `aula28` DEFAULT CHARACTER SET utf8 ;

CREATE TABLE IF NOT EXISTS `aula28`.`Area` ( `codigo_regiao` VARCHAR(2) NOT NULL, `tipo_regiao` VARCHAR(25) NULL DEFAULT NULL, `nome` VARCHAR(30) NULL DEFAULT NULL, `Empregado_codigo_empregado` INT(11) NOT NULL, `Area_codigo_regiao` VARCHAR(2) NOT NULL, PRIMARY KEY (`codigo_regiao`), INDEX `fk_Area_Empregado_idx` (`Empregado_codigo_empregado` ASC) VISIBLE, INDEX `fk_Area_Area1_idx` (`Area_codigo_regiao` ASC) VISIBLE, CONSTRAINT `fk_Area_Empregado` FOREIGN KEY (`Empregado_codigo_empregado`) REFERENCES `aula28`.`Empregado` (`codigo_empregado`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `fk_Area_Area1` FOREIGN KEY (`Area_codigo_regiao`) REFERENCES `aula28`.`Area` (`codigo_regiao`) ON DELETE NO ACTION ON UPDATE NO ACTION)ENGINE = InnoDBDEFAULT CHARACTER SET = utf8;

CREATE TABLE IF NOT EXISTS `aula28`.`Empregado` ( `codigo_empregado` INT(11) NOT NULL, `tipo_empregado` VARCHAR(15) NULL DEFAULT NULL, `nome` VARCHAR(30) NULL DEFAULT NULL, `cota` VARCHAR(9) NULL DEFAULT NULL, PRIMARY KEY (`codigo_empregado`))ENGINE = InnoDBDEFAULT CHARACTER SET = utf8;

SET SQL_MODE=@OLD_SQL_MODE;SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Banco Criado







- Selecione uma das tabelas
- E clique em Select Rows

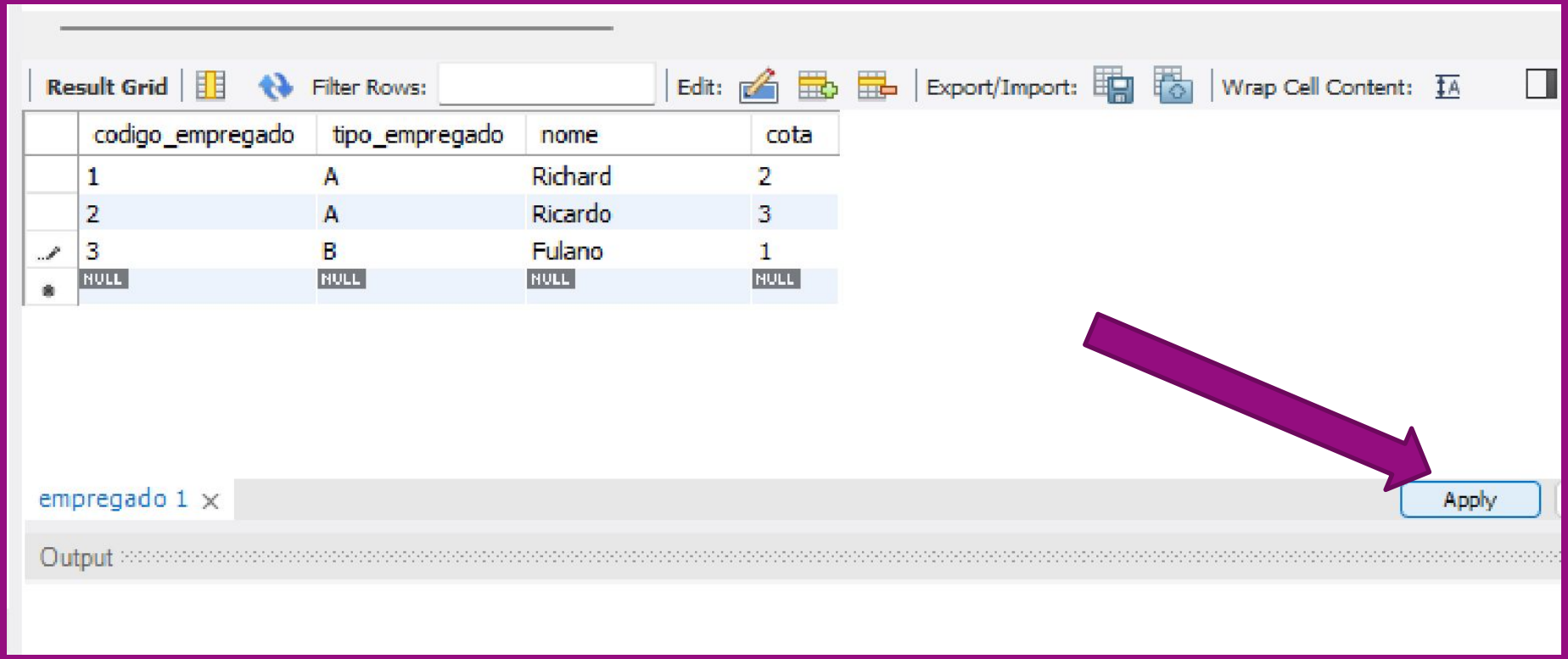
SQL
Structure Query Language

- INSERT INTO <nome da tabela>

  (<nome da(s) coluna(s)>)

  VALUES (<valores>);

Vamos Preencher os dados

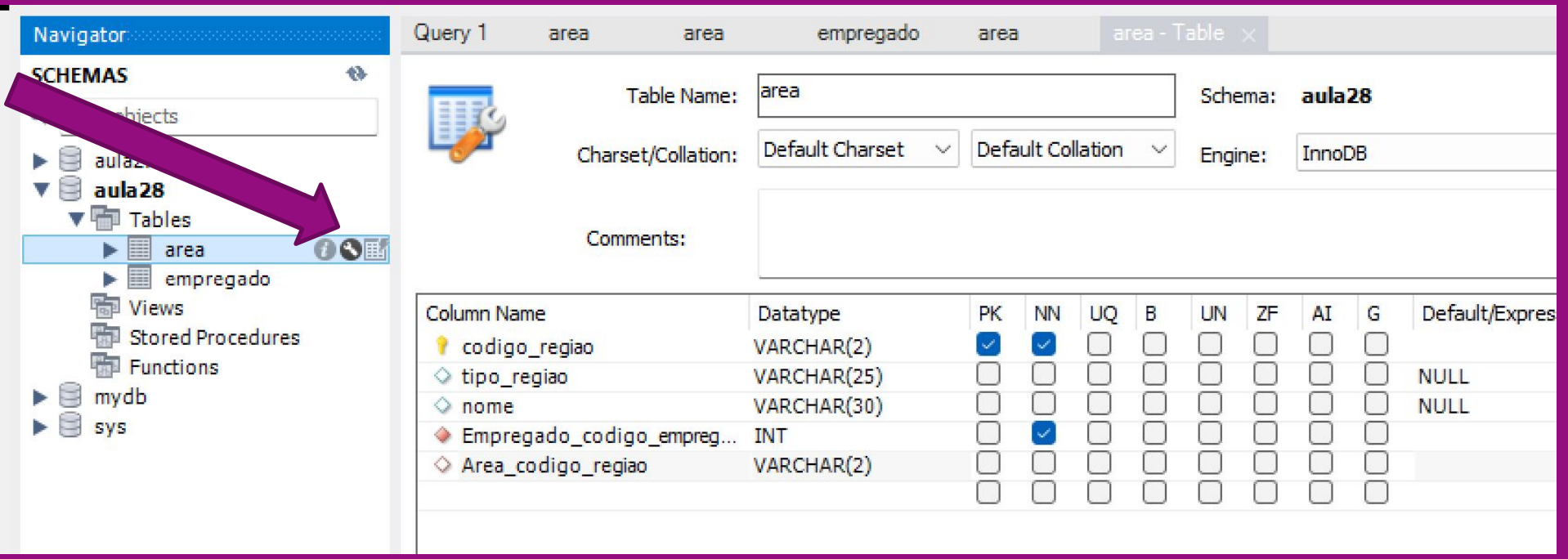


A ferramenta gerou os Inserts

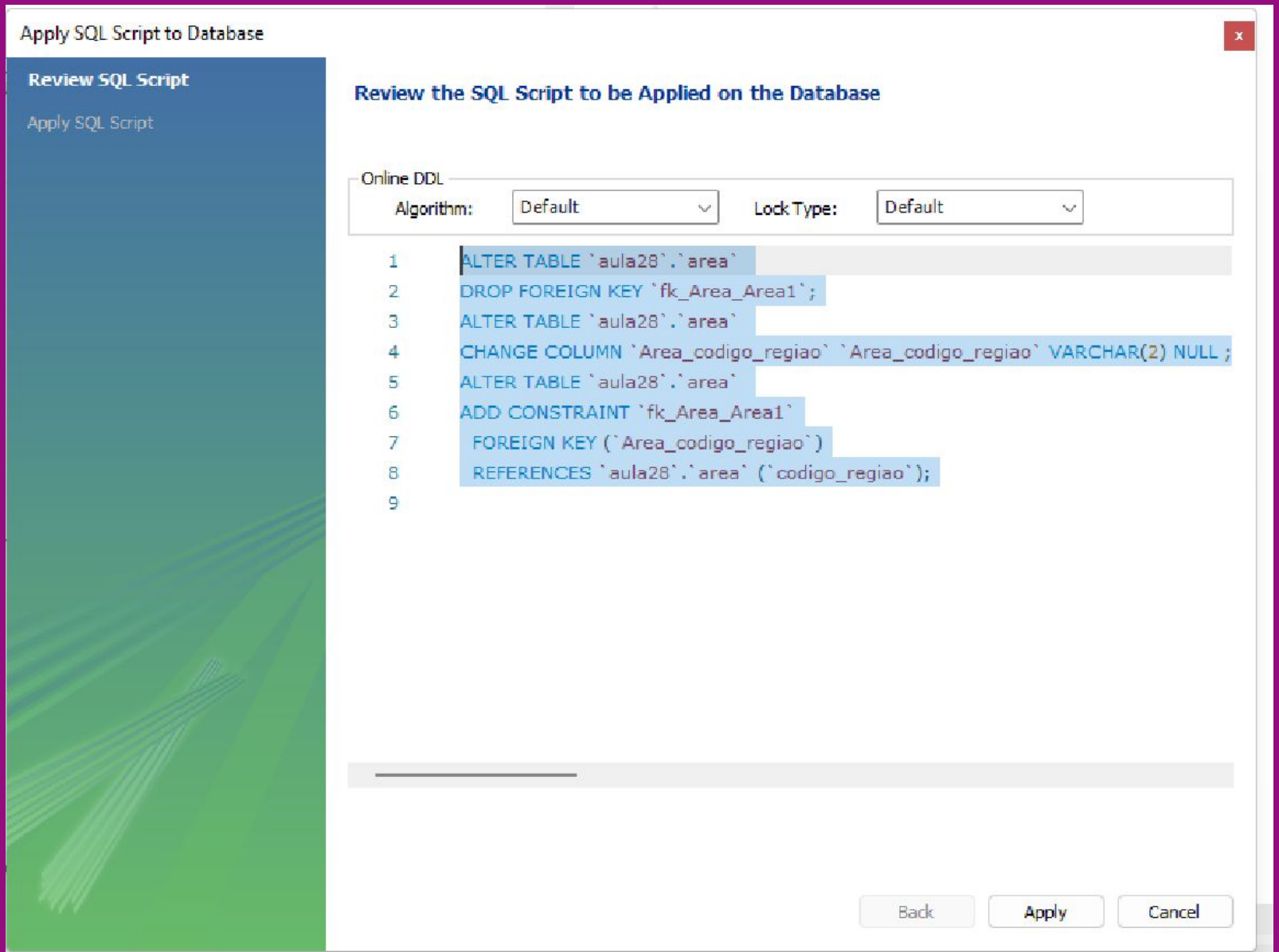
```
INSERT INTO `aula28`.`empregado` (`codigo_empregado`, `tipo_empregado`, `nome`, `cota`) VALUES ('1', 'A', 'Richard', '2’);
INSERT INTO `aula28`.`empregado` (`codigo_empregado`, `tipo_empregado`, `nome`, `cota`) VALUES ('2', 'A', 'Ricardo', '3’);
INSERT INTO `aula28`.`empregado` (`codigo_empregado`, `tipo_empregado`, `nome`, `cota`) VALUES ('3', 'B', 'Fulano', '1');
```

Botão direito na chave

- Auto referencia: Vamos deixar que possa ser null
- Depois clique em apply



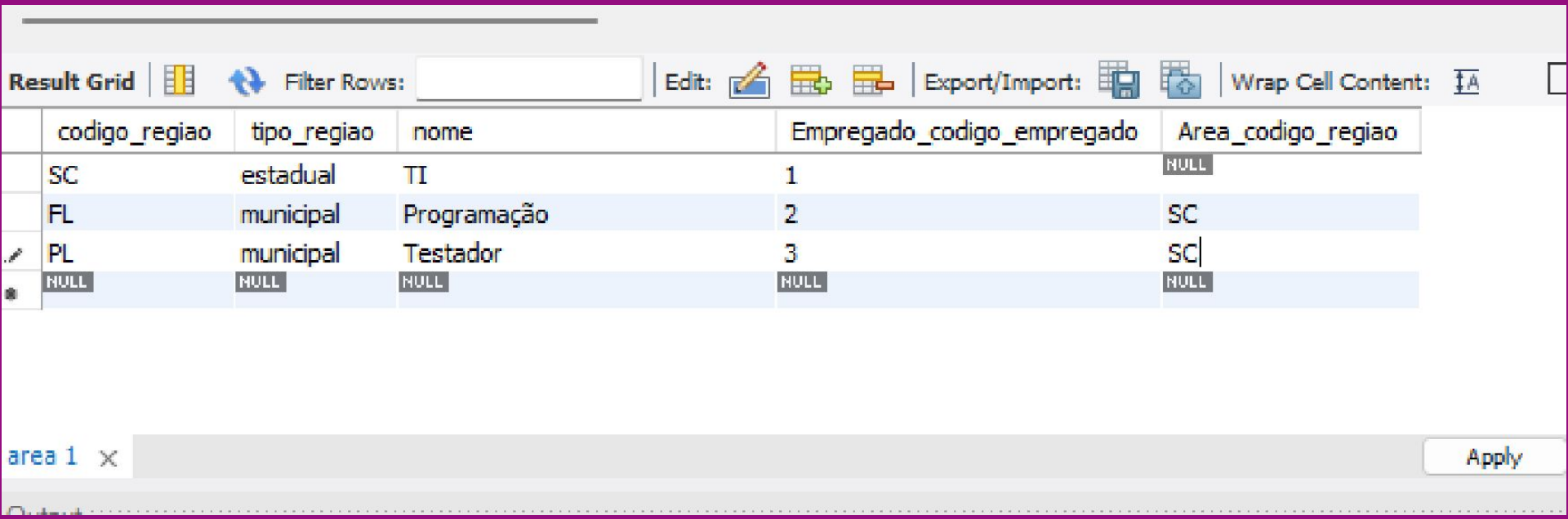
Alter Table


Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Review the SQL Script to be Applied on the Database
Online DDL
Algorithm:
Default
Lock Type:
Default
1 ALTER TABLE `aula28`.`area`
2 DROP FOREIGN KEY `fk_Area_Area1`;
3 ALTER TABLE `aula28`.`area`
4 CHANGE COLUMN `Area_codigo_regiao` `Area_codigo_regiao` VARCHAR(2) NULL ;
5 ALTER TABLE `aula28`.`area`
6 ADD CONSTRAINT `fk_Area_Area1`
7 FOREIGN KEY (`Area_codigo_regiao`)
8 REFERENCES `aula28`.`area` (`codigo_regiao`);
9
Back
Apply
Cancel

Alter Table

- ALTER TABLE `aula28`.`area` DROP FOREIGN KEY `fk_Area_Area1`;
- ALTER TABLE `aula28`.`area` CHANGE COLUMN `Area_codigo_regiao` `Area_codigo_regiao` VARCHAR(2) NULL ;ALTER TABLE `aula28`.`area` ADD CONSTRAINT `fk_Area_Area1` FOREIGN KEY (`Area_codigo_regiao`) REFERENCES `aula28`.`area` (`codigo_regiao`);

Agora Podemos preencher



INSERT

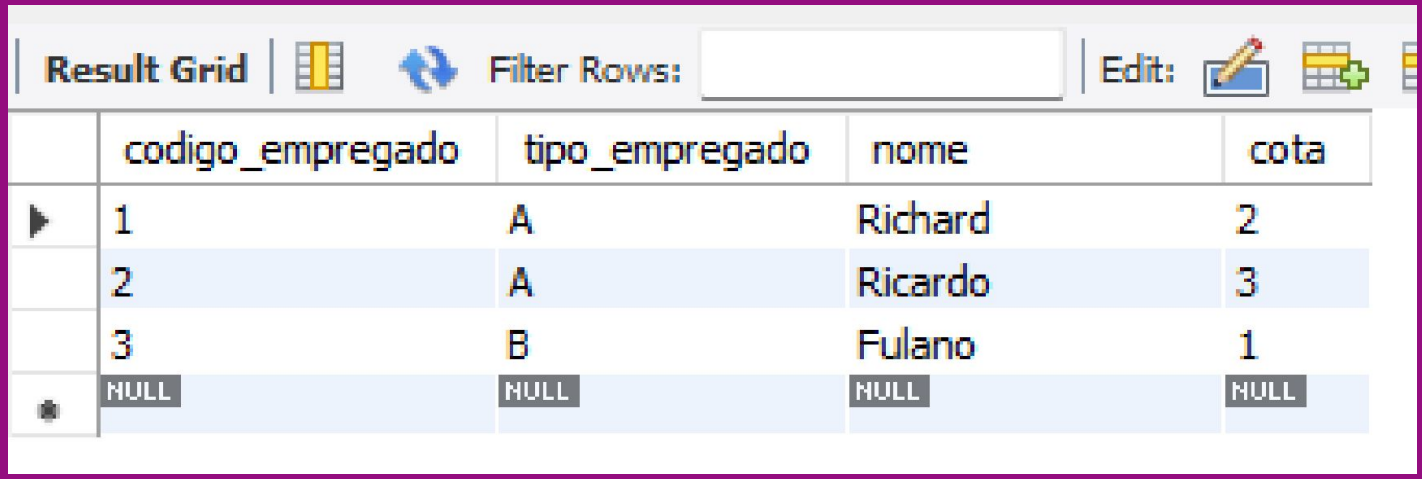
- INSERT INTO `aula28`.`area` (`codigo_regiao`, `tipo_regiao`, `nome`, `Empregado_codigo_empregado`) VALUES ('SC', 'estadual', 'TI', '1’);
- INSERT INTO `aula28`.`area` (`codigo_regiao`, `tipo_regiao`, `nome`, `Empregado_codigo_empregado`, `Area_codigo_regiao`) VALUES ('FL', 'municipal', 'Programação', '2', 'SC’);
- INSERT INTO `aula28`.`area` (`codigo_regiao`, `tipo_regiao`, `nome`, `Empregado_codigo_empregado`, `Area_codigo_regiao`) VALUES ('PL', 'municipal', 'Testador', '3', 'SC');

SQL
Structure Query Language

- SELECT: especifica as colunas da tabela que queremos selecionar.
- FROM: especifica as tabelas.
- WHERE: especifica a condição de seleção das linhas.

Exemplo

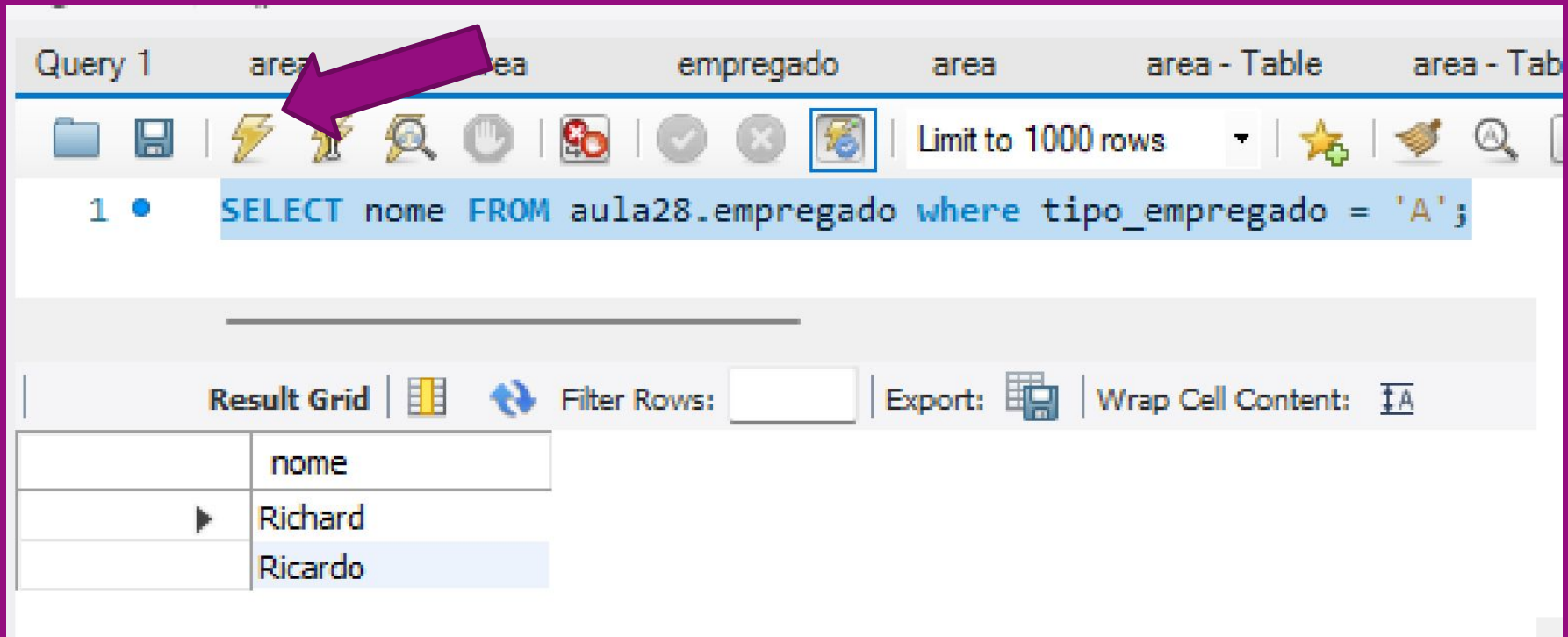
- SELECT * FROM aula28.empregado;
- Retorna todos



codigo_empregado	tipo_empregado	nome	cota
1	A	Richard	2
2	A	Ricardo	3
3	B	Fulano	1
NULL	NULL	NULL	NULL

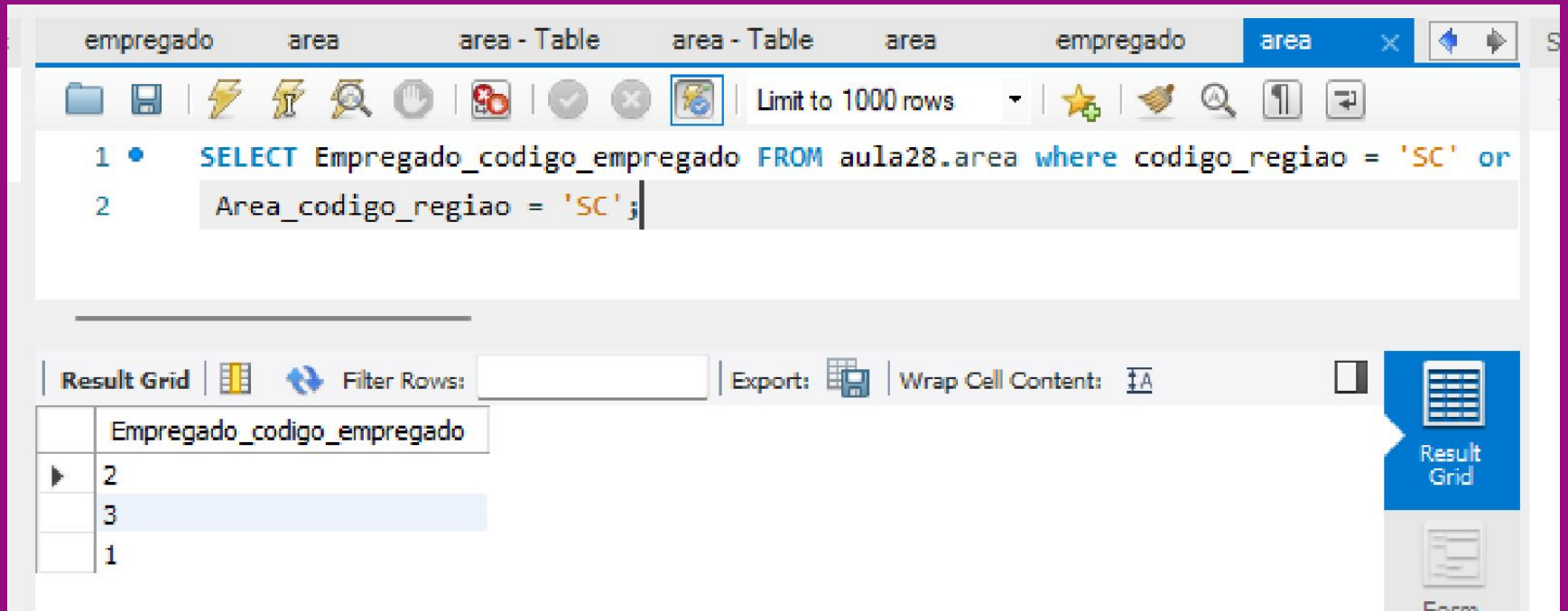
Apenas os nomes dos empregados tipo A

- SELECT nome FROM aula28.empregado where tipo_empregado = 'A';



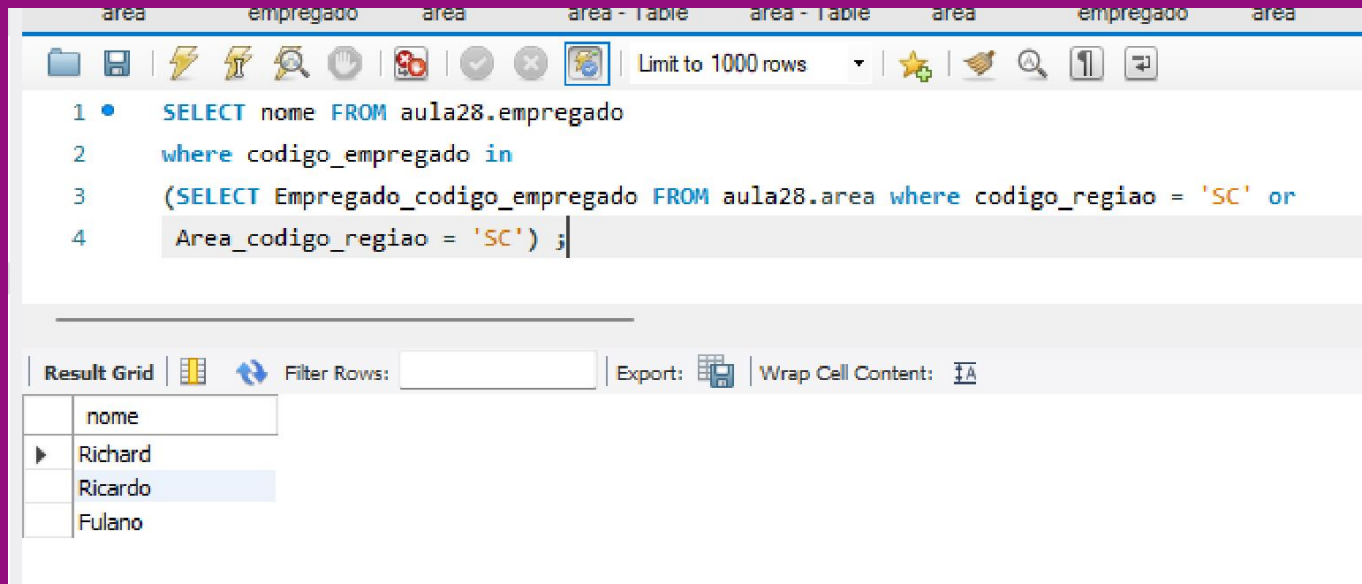
Códigos dos empregados de SC

- SELECT Empregado_codigo_empregado FROM aula28.area where codigo_regiao = 'SC' or Area_codigo_regiao = 'SC';



SELECT dentro de SELECT

- SELECT nome FROM aula28.empregado where codigo_empregado in (SELECT Empregado_codigo_empregado FROM aula28.area where codigo_regiao = 'SC' or Area_codigo_regiao = 'SC') ;



Exercício 2

- Faça a consulta dos empregados que contenham a string ‘Ric’ no nome

- Faça a consulta dos empregados que contenham cotas abaixo ou igual a 2

Exercício 3

A. Faça a consulta das áreas que contenham a letra T
B. Faça a consulta das áreas que são municipais

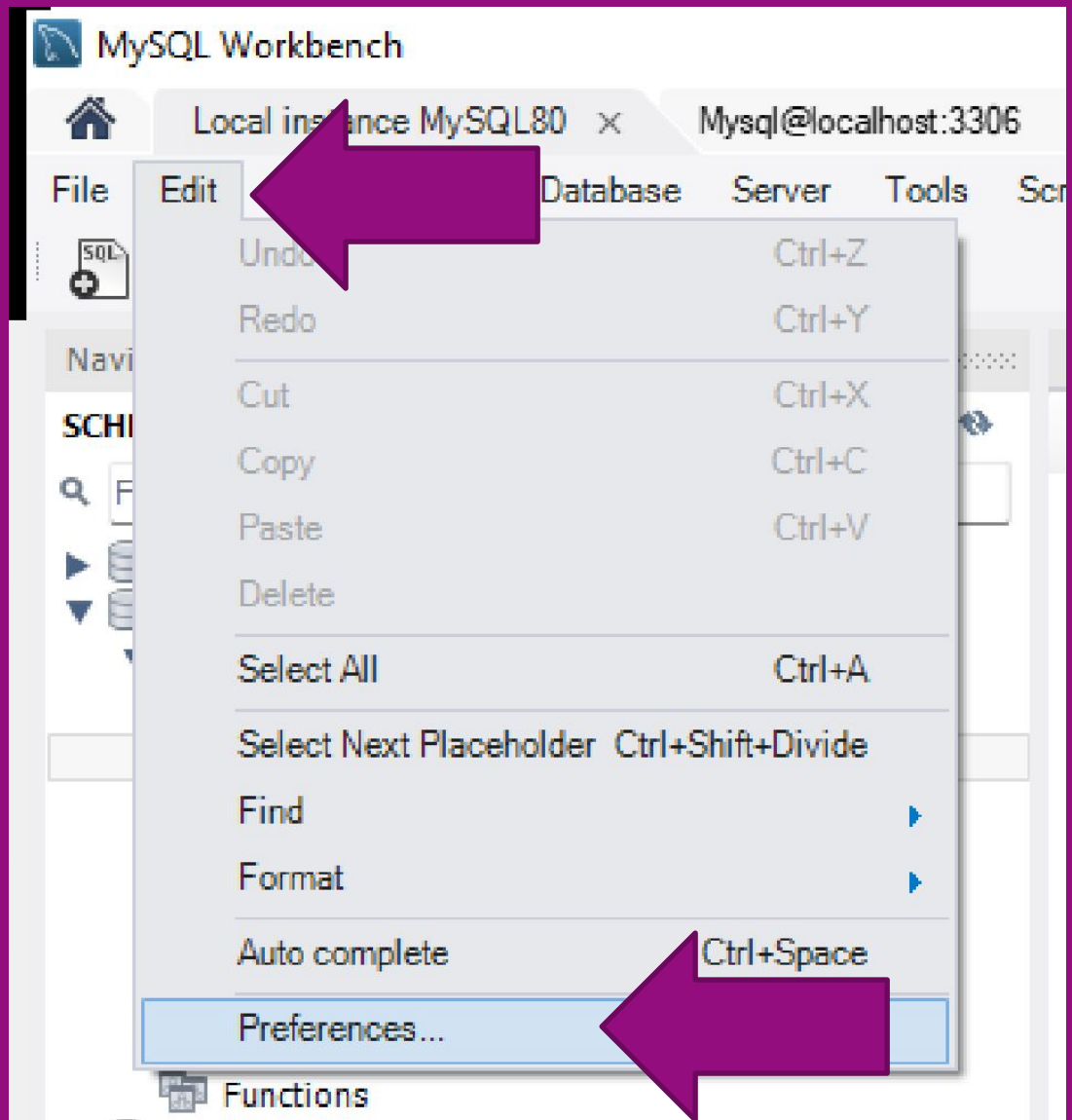
Alterar

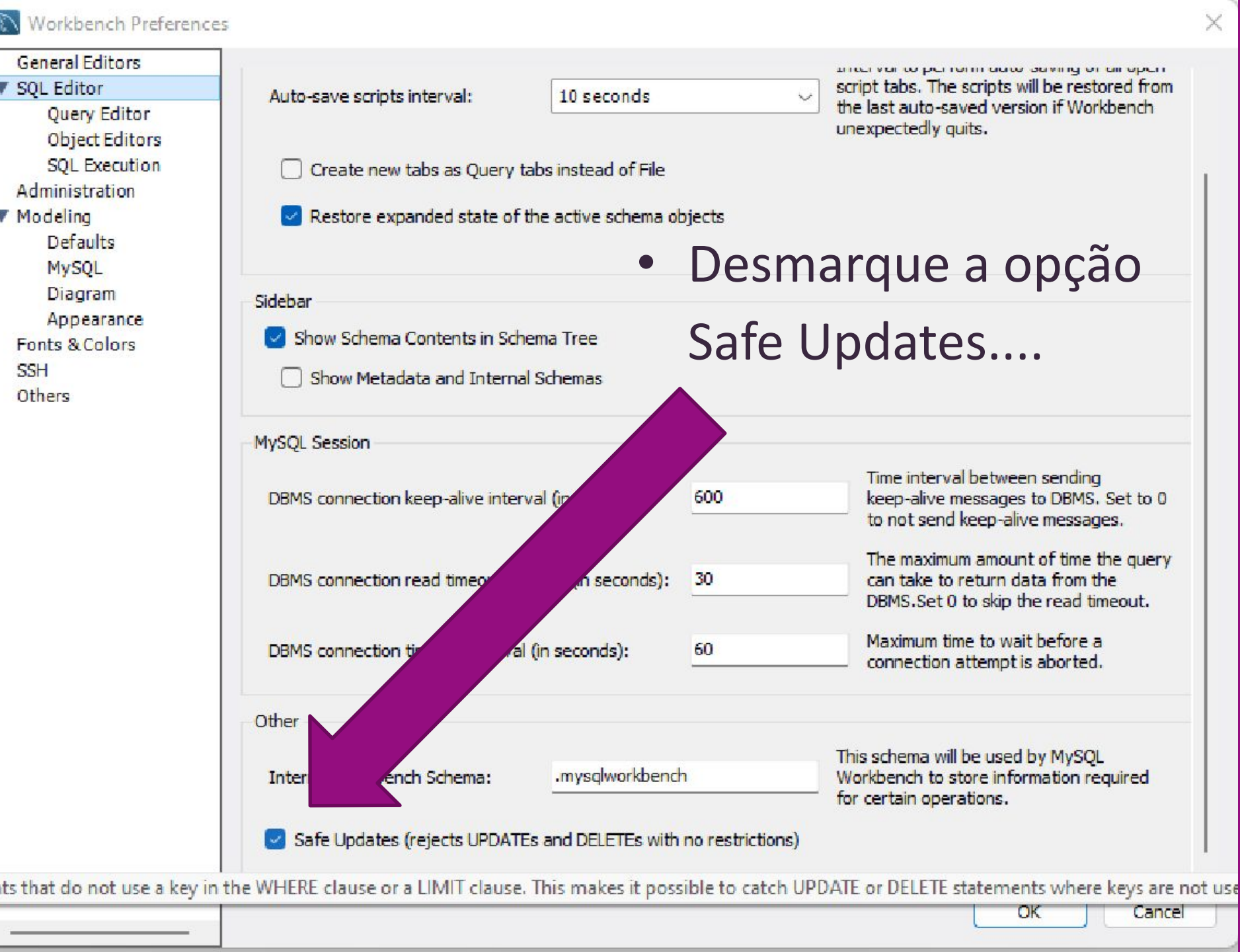
- UPDATE <nome da tabela>

  SET <nome da(s) coluna(s)> = valor

  WHERE <condição>

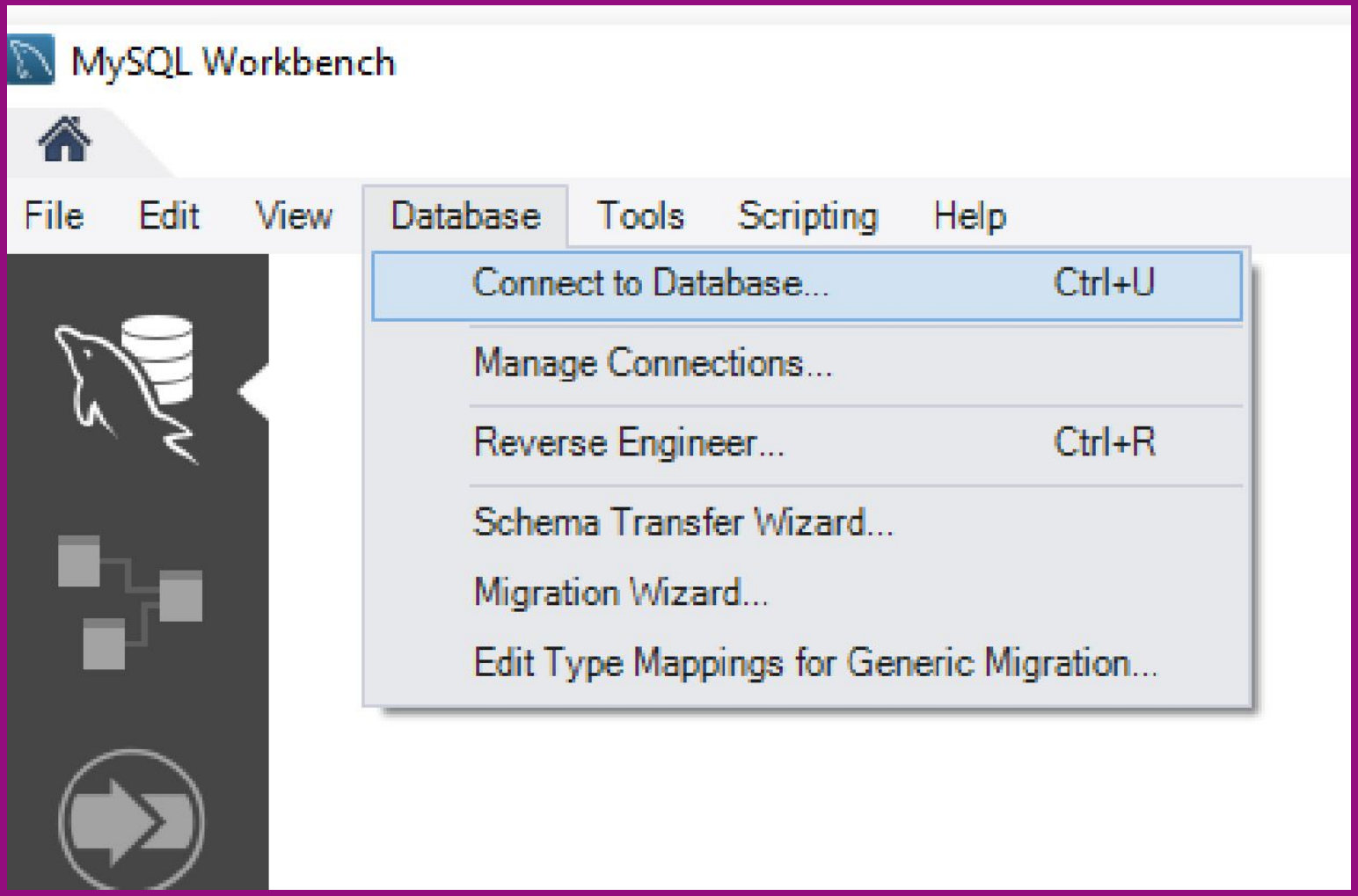
Mas antes de testar
Vamos tirar desmarcar o Safe mode





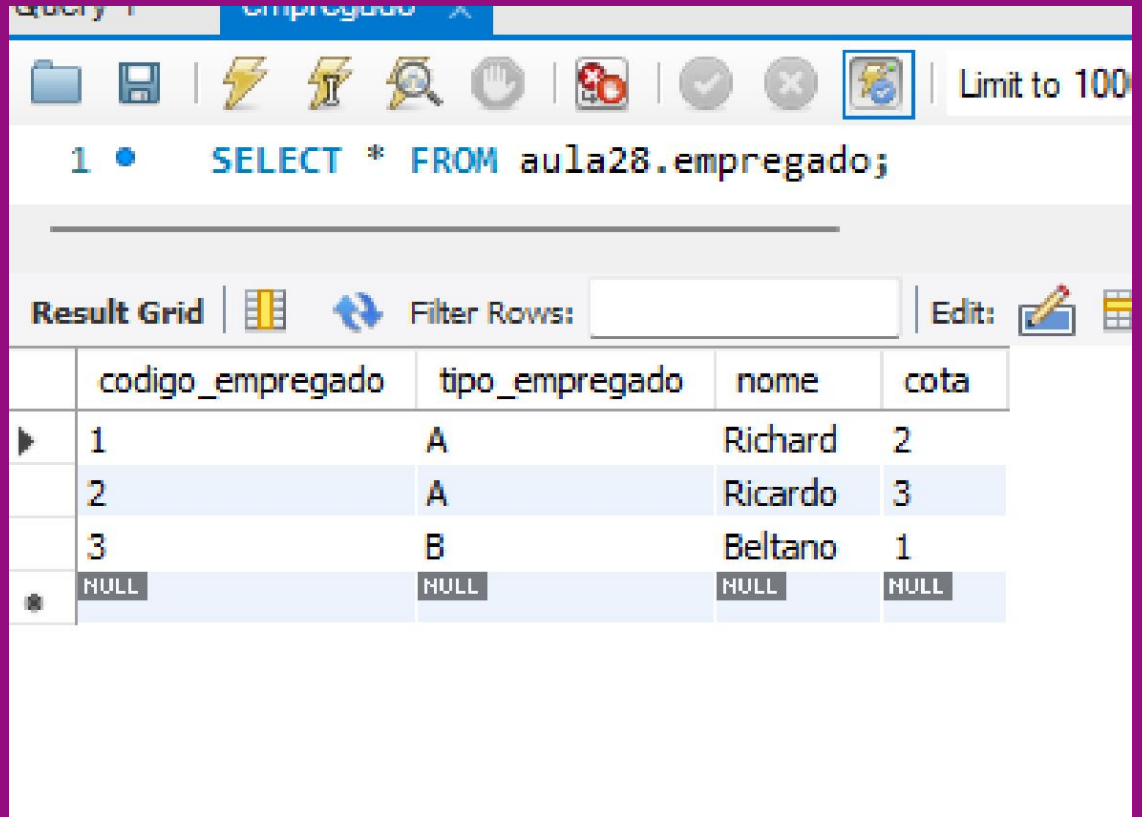
- Desmarque a opção Safe Updates....

Reconect ao banco



UPDATE

- UPDATE aula28.empregado SET nome = 'Beltano' where nome = 'Fulano';



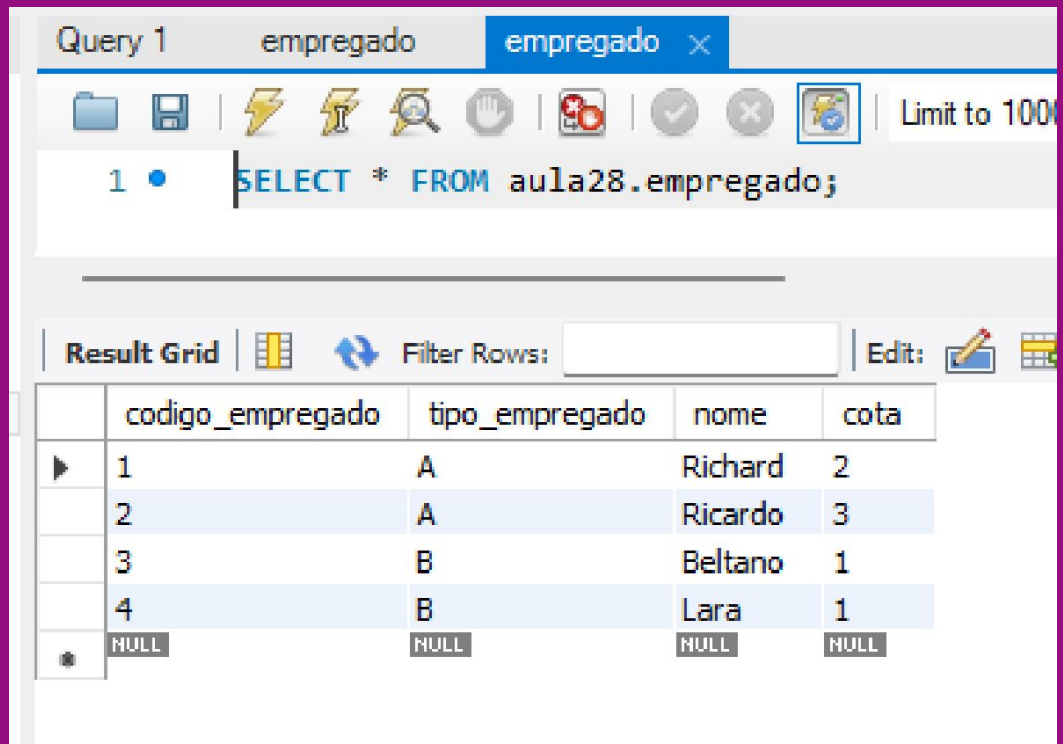
Lembrando do INSERT

- INSERT INTO <nome da tabela> (<nome da(s) coluna(s)>)

  VALUES (<valores>);

Inserindo novo registro

- INSERT INTO aula28.empregado (codigo_empregado, tipo_empregado , nome , cota) VALUES (4, 'B', 'Lara', 1);



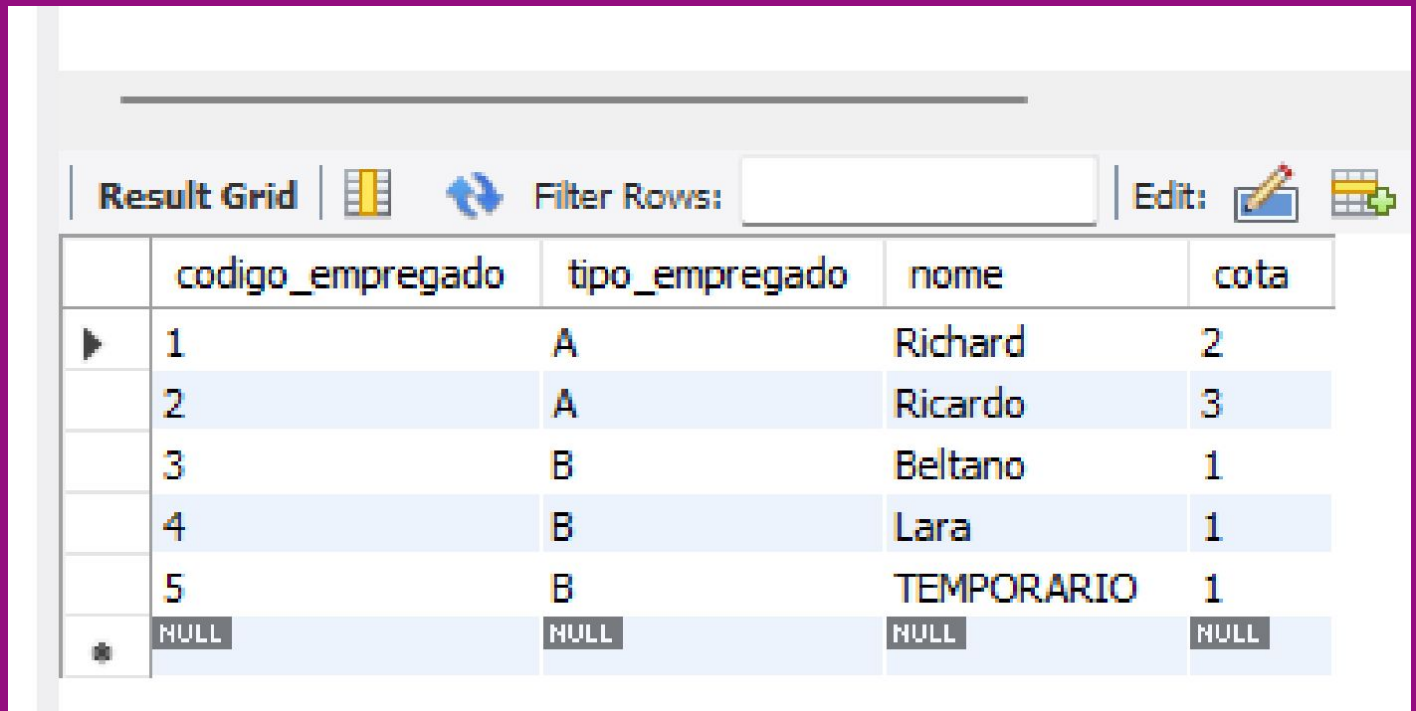
Exercício 4

- Adicione um Registro na tabela AREA para o código do novo empregado.

- Mude o nome do empregado Beltrano para a o seu nome.

VAMOS INSERIR +1

- INSERT INTO aula28.empregado (codigo_empregado, tipo_empregado , nome , cota) VALUES (5, 'B', 'TEMPORARIO', 1);



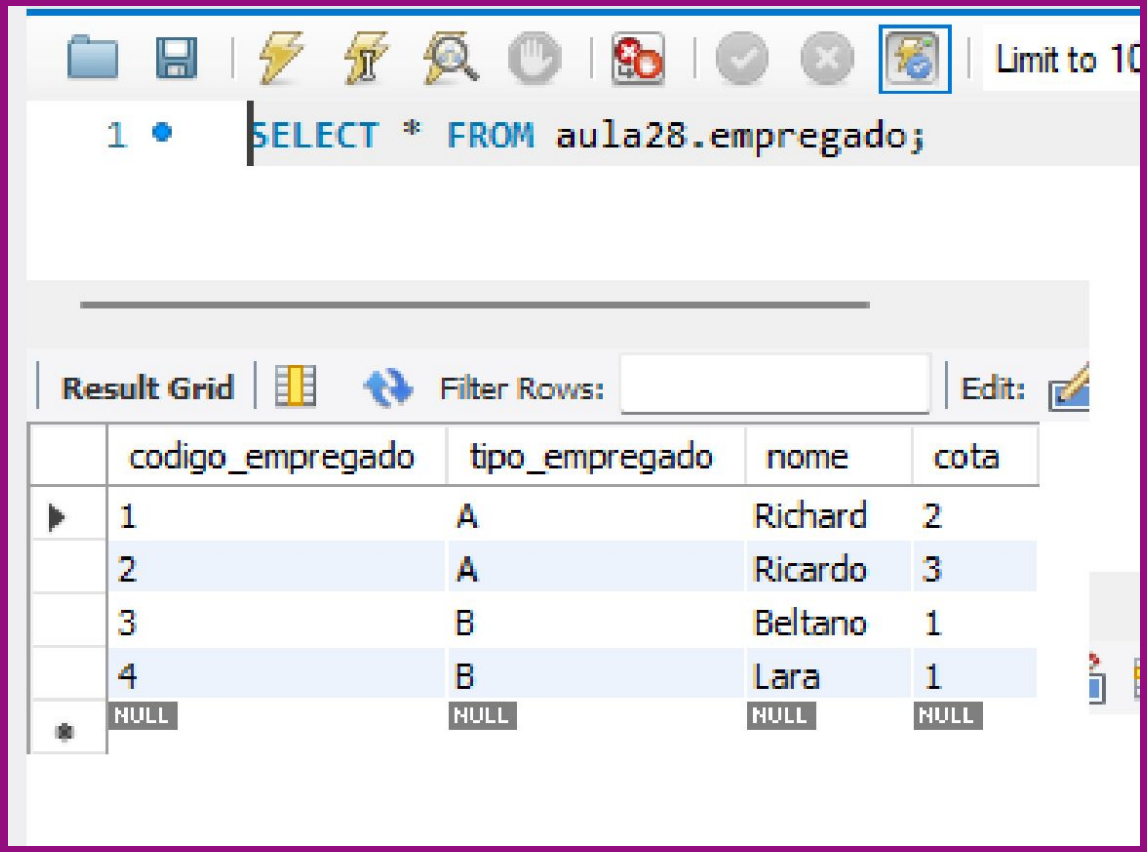
codigo_empregado	tipo_empregado	nome	cota
1	A	Richard	2
2	A	Ricardo	3
3	B	Beltano	1
4	B	Lara	1
5	B	TEMPORARIO	1
NULL	NULL	NULL	NULL

DELETE

- DELETE FROM <nome da tabela> WHERE <condição>;

DELETE

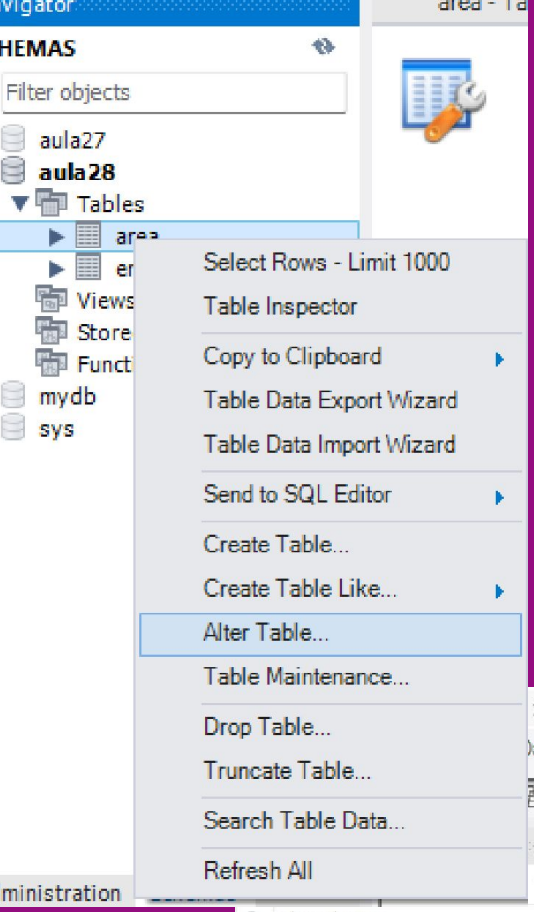
- DELETE FROM aula28.empregado where codigo_empregado = 5;

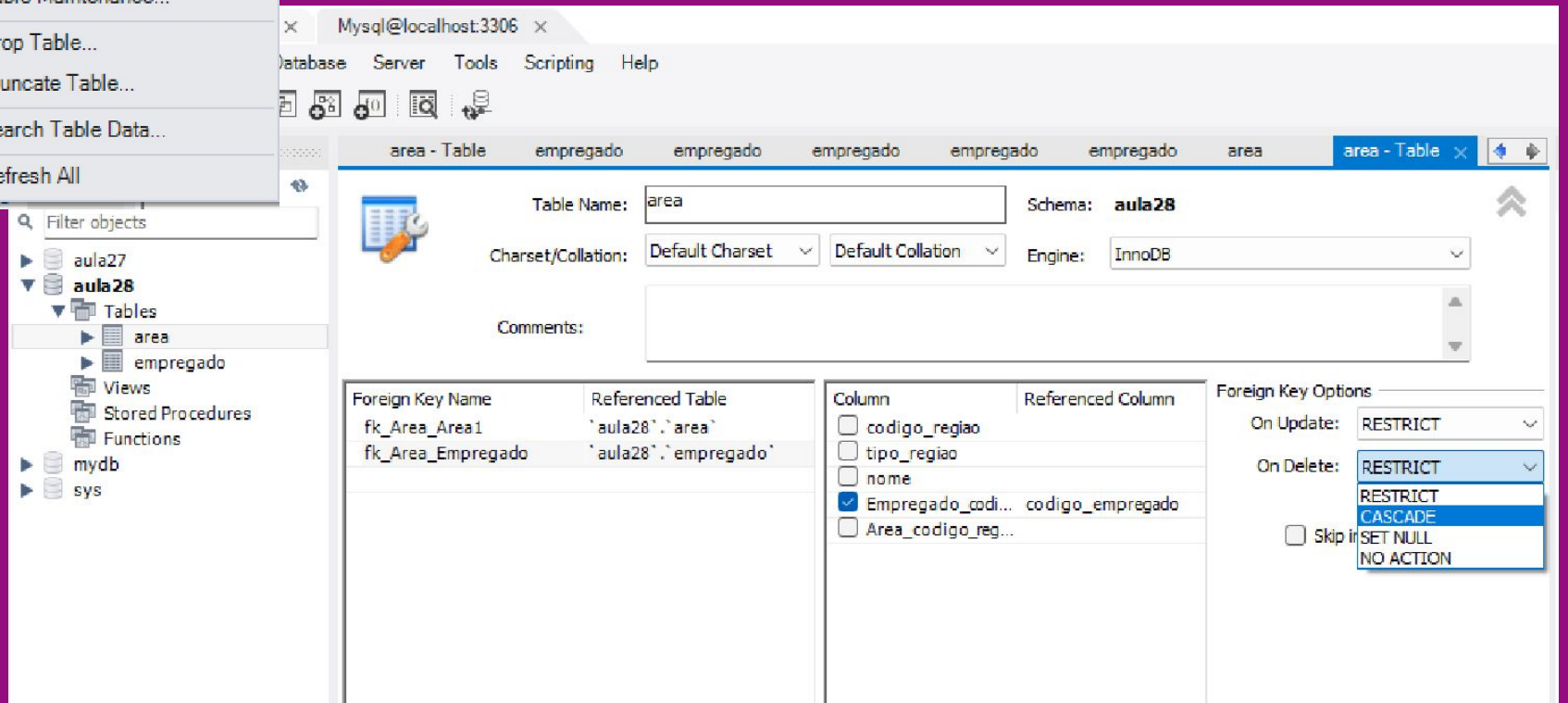


O que acontece ?

- DELETE FROM aula28.empregado where codigo_empregado = 2;
- Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`aula28`.`area`, CONSTRAINT `fk_Area_Empregado` FOREIGN KEY (`Empregado_codigo_empregado`) REFERENCES `empregado` (`codigo_empregado`)) 0.000 sec

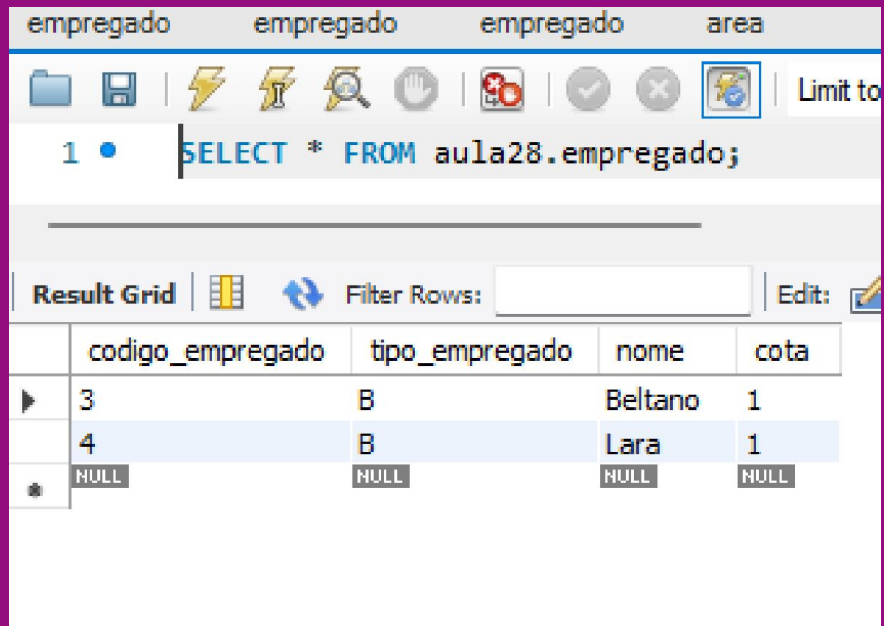
Alterar a TABELA





O que acontece ?

- DELETE FROM aula28.empregado where codigo_empregado = 2;
- DELETE FROM aula28.empregado where codigo_empregado = 1;



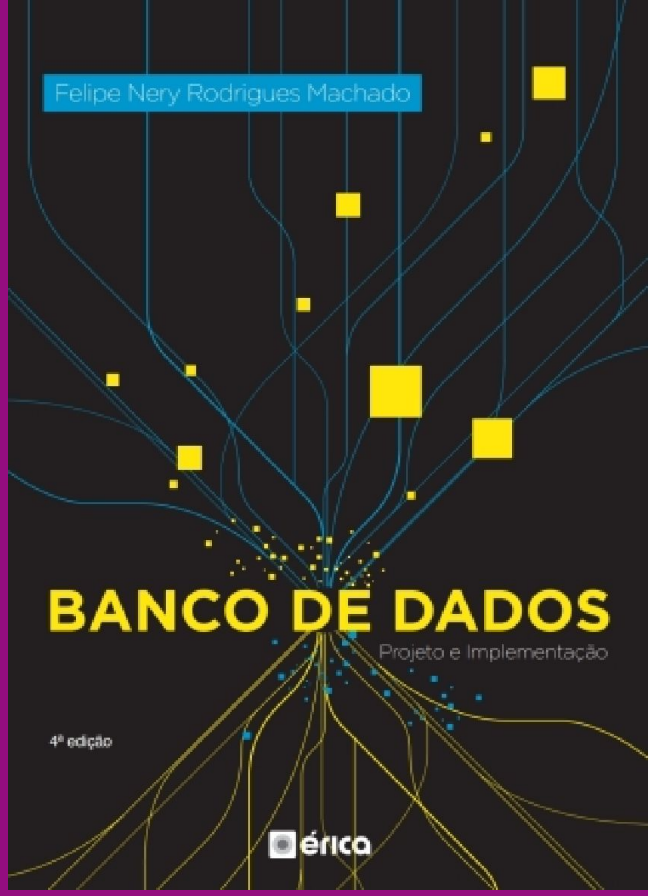
Exercício 5

- Apage o registro da Area SC
  - Lembre-se tem que alterar a estrutura da tabela por causa da foreign key

Presença

- Faça os exercícios e responda o formulário que será passado no chat,
- Prazo: até dia 05/06

Busca ATIVA



- Leitura e execução dos exemplos livro:
  - Machado, Felipe Nery R. *BANCO DE DADOS – PROJETO E IMPLEMENTAÇÃO*. Disponível em: Minha Biblioteca, (4th edição). Editora Saraiva, 2020.
- Principalmente, antes da A2

Referências bibliográficas

Machado, Felipe Nery R. BANCO DE DADOS – PROJETO E IMPLEMENTAÇÃO. Disponível em: Minha Biblioteca, (4th edição). Editora Saraiva, 2020.

DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2004. 865 p. ISBN 85-352-1273-6.

ELMASRI, Ramez; NAVATHE, Sham. **Sistemas de banco de dados**. 4. ed. São Paulo: Addison-Wesley, 2005. 724 p. ISBN 8588639173.

NAVATHE, Elmasri. **Sistema de Banco de Dados**. 6ª Edição. São Paulo: Person Addison Wesley, 2011.

PEREIRA, Silvio do Lago. **Estruturas de dados fundamentais: conceitos e aplicações**. 8. ed. São Paulo: Érica, 2004. 238 p. ISBN 85-7194-370-2.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed. São Paulo: Makron Books, 2004. 778 p. ISBN 85-346-1073-8.

CRÉDITOS

COORDENAÇÃO

Vera Rejane Niedersberg Schuhmacher

PROFESSORES

Rafael Lessa
Daniella Vieira