



INSTITUTO FEDERAL  
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
Bahia

---

# Banco de Dados

Gerando o Modelo Físico a partir do  
Modelo Relacional (Lógico)



# Roteiro

- Gerando o Modelo Físico a partir do Modelo Relacional (Lógico):
  - Componentes do Modelo Lógico;
  - Componentes do Modelo Físico;
  - Do Modelo Lógico para o Modelo Físico;
  - Executar o Script SQL de criação de BD;
  - Explicação do Script SQL de criação de BD;
  - Atividades.



# Componentes do Modelo Lógico

---

## Modelo Relacional Lógico:

- Estrutura de tabelas;
- Colunas;
- Tipos de dados;
- Chaves primárias;
- Chaves estrangeiras;
- Restrições de integridade.



# Componentes do Modelo Físico

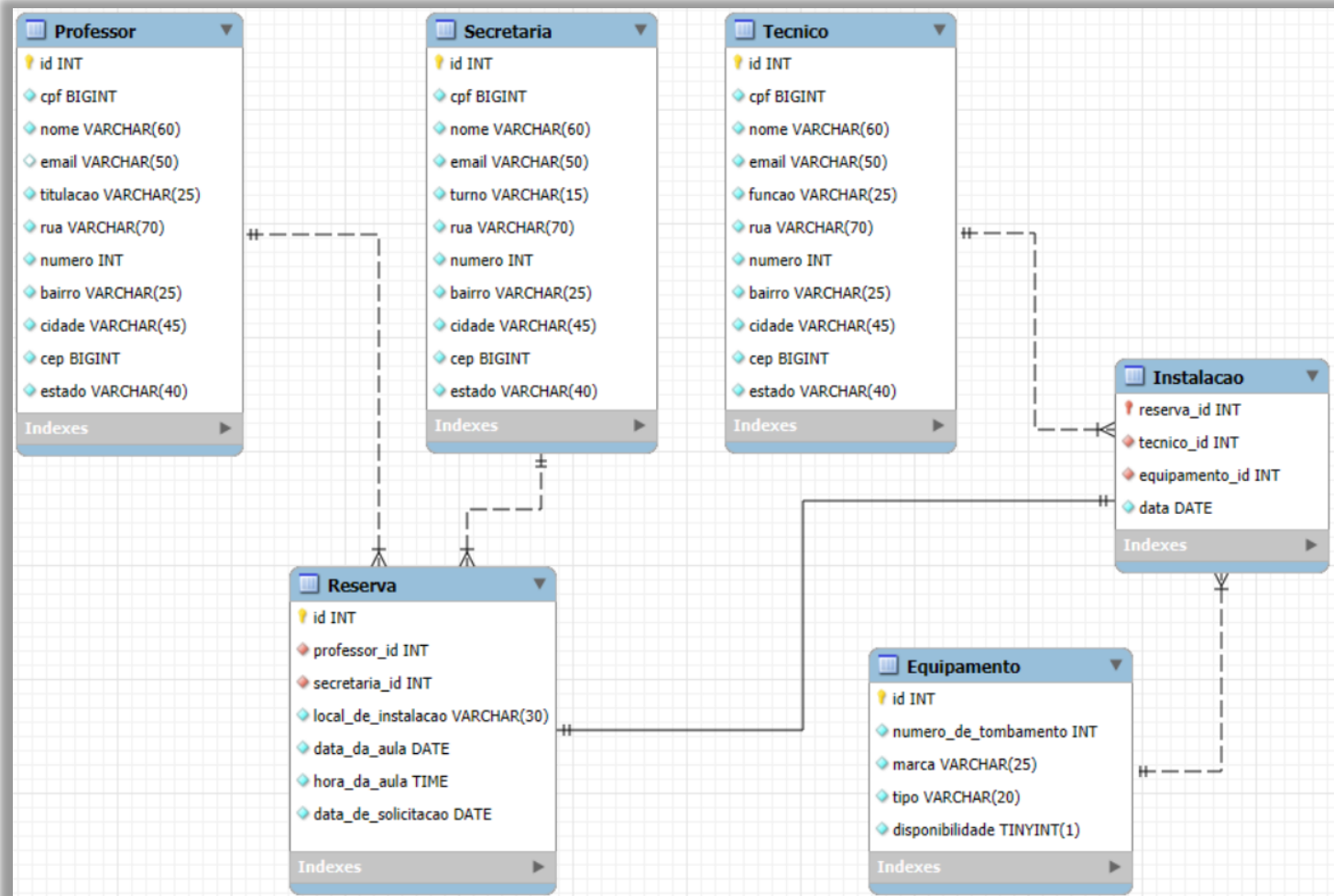
---

## Modelo Físico:

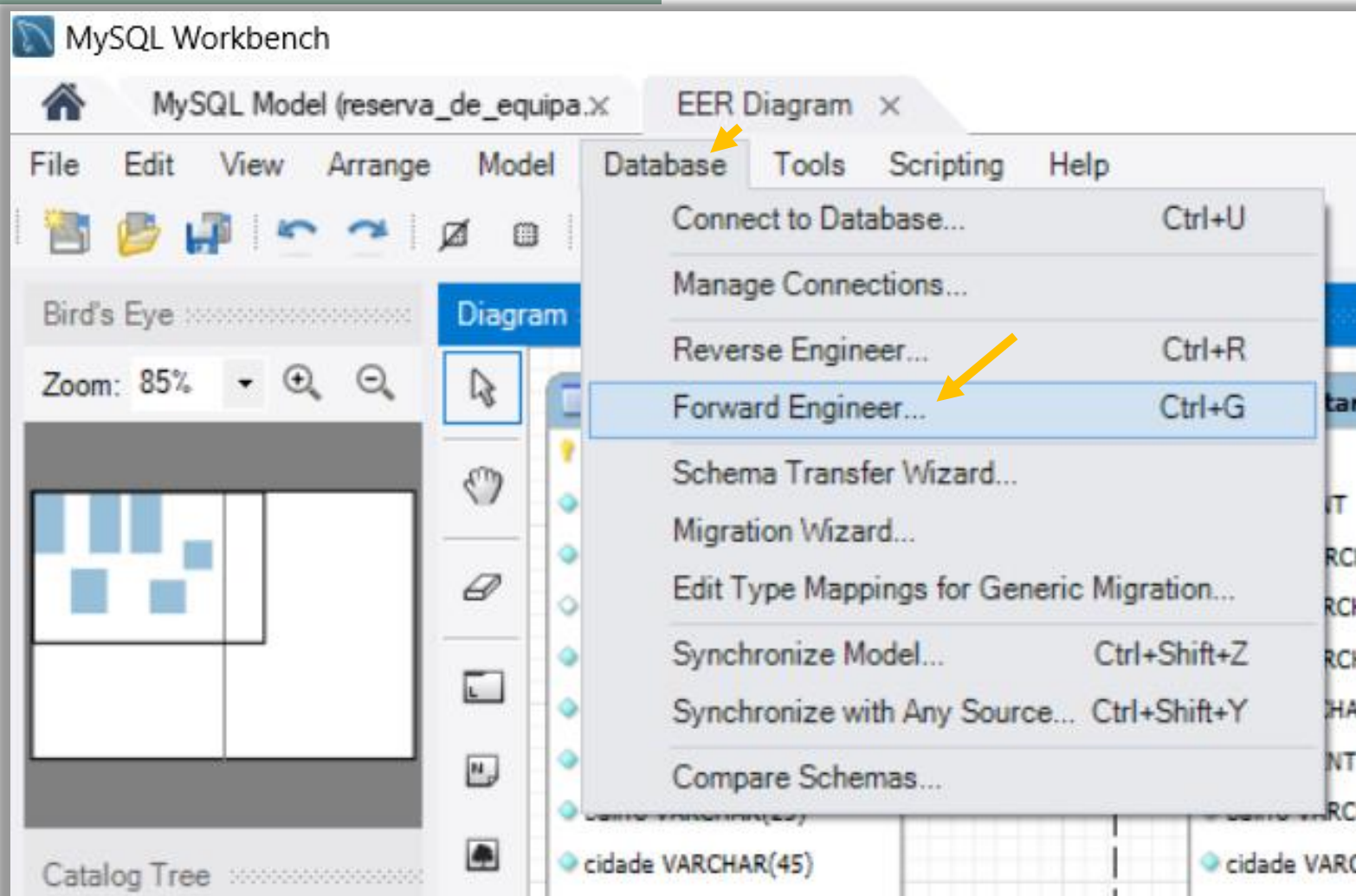
- Estrutura de armazenamento em disco;
- Índices;
- Particionamento de tabelas;
- Estratégias de otimização de desempenho.



# Do Modelo Lógico para o Modelo Físico



# Do Modelo Lógico para o Modelo Físico



# Do Modelo Lógico para o Modelo Físico

Stored Connection:  Select from saved connection settings

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:   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.



# Do Modelo Lógico para o Modelo Físico

## Tables

- ☐ Skip creation of FOREIGN KEYS
- ☒ Skip creation of FK Indexes as well
- ☐ Generate separate CREATE INDEX statements
- ☐ Generate INSERT statements for tables
- ☐ Disable FK checks for INSERTs

## Other Objects

- ☒ Don't create view placeholder tables
- ☒ Do not create users. Only create privileges (GRANTs)

## Code Generation

- ☐ DROP objects before each CREATE object
- ☐ Generate DROP SCHEMA
- ☒ Omit schema qualifier in object names
- ☒ Generate USE statements
- ☐ Add SHOW WARNINGS after every DDL statement
- ☐ Include model attached scripts

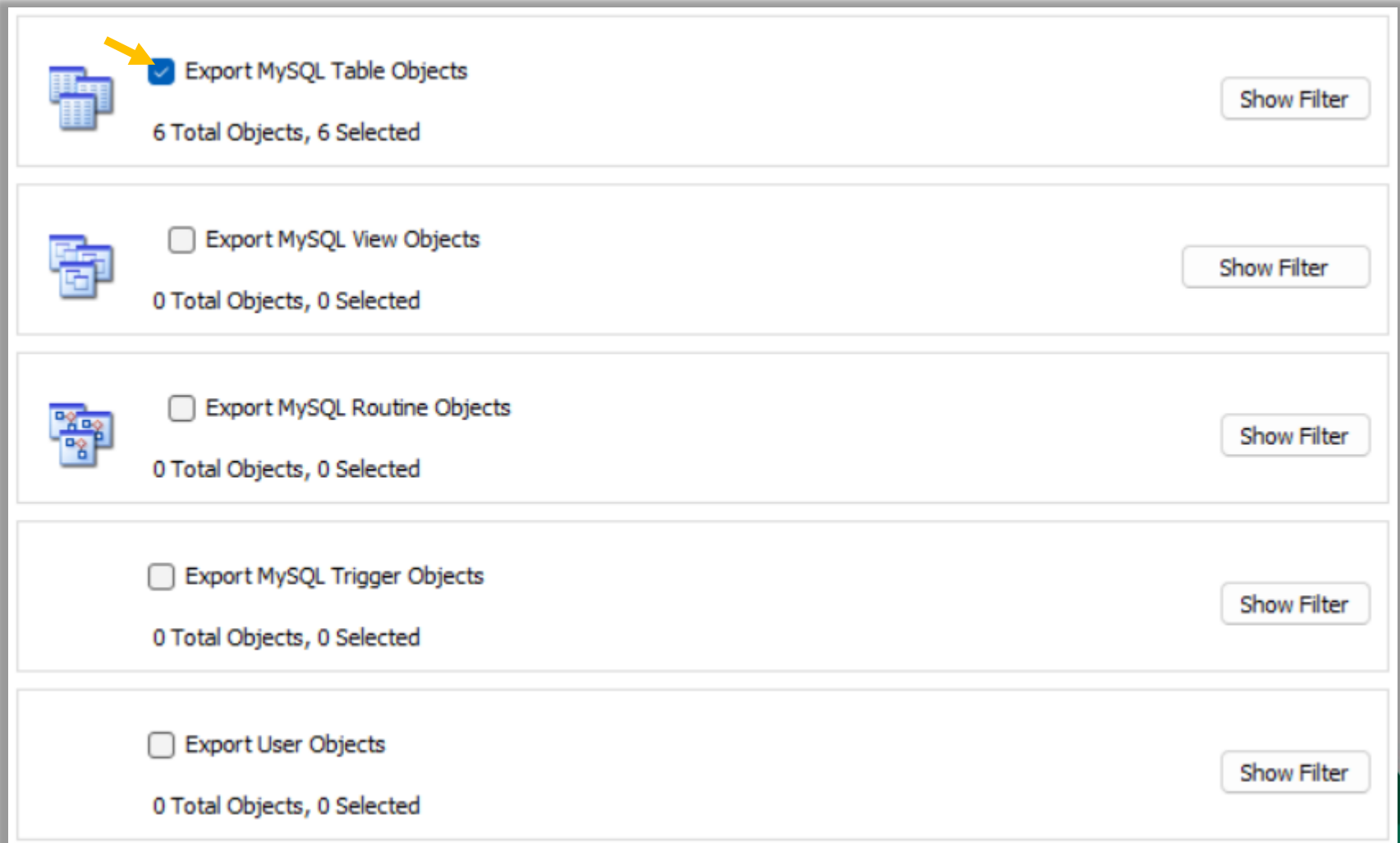
Back

→ Next

Cancel



# Do Modelo Lógico para o Modelo Físico



☒ Export MySQL Table Objects  
6 Total Objects, 6 Selected [Show Filter](#)

☐ Export MySQL View Objects  
0 Total Objects, 0 Selected [Show Filter](#)

☐ Export MySQL Routine Objects  
0 Total Objects, 0 Selected [Show Filter](#)

☐ Export MySQL Trigger Objects  
0 Total Objects, 0 Selected [Show Filter](#)

☐ Export User Objects  
0 Total Objects, 0 Selected [Show Filter](#)



# Do Modelo Lógico para o Modelo Físico

```
11  -----
12  -- Schema reserva_de Equipamentos
13  -----
14  CREATE SCHEMA IF NOT EXISTS `reserva_de Equipamentos` DEFAULT CHARACTER SET utf8 ;
15  USE `reserva_de Equipamentos` ;
16
17  -----
18  -- Table `Professor`
19  -----
20  CREATE TABLE IF NOT EXISTS `Professor` (
21    `id` INT NOT NULL AUTO_INCREMENT,
22    `cpf` BIGINT NOT NULL,
23    `nome` VARCHAR(60) NOT NULL,
24    `email` VARCHAR(50) NULL,
```

Script SQL Gerado

Save to File...

Copy to Clipboard

Salvar Script SQL Gerado

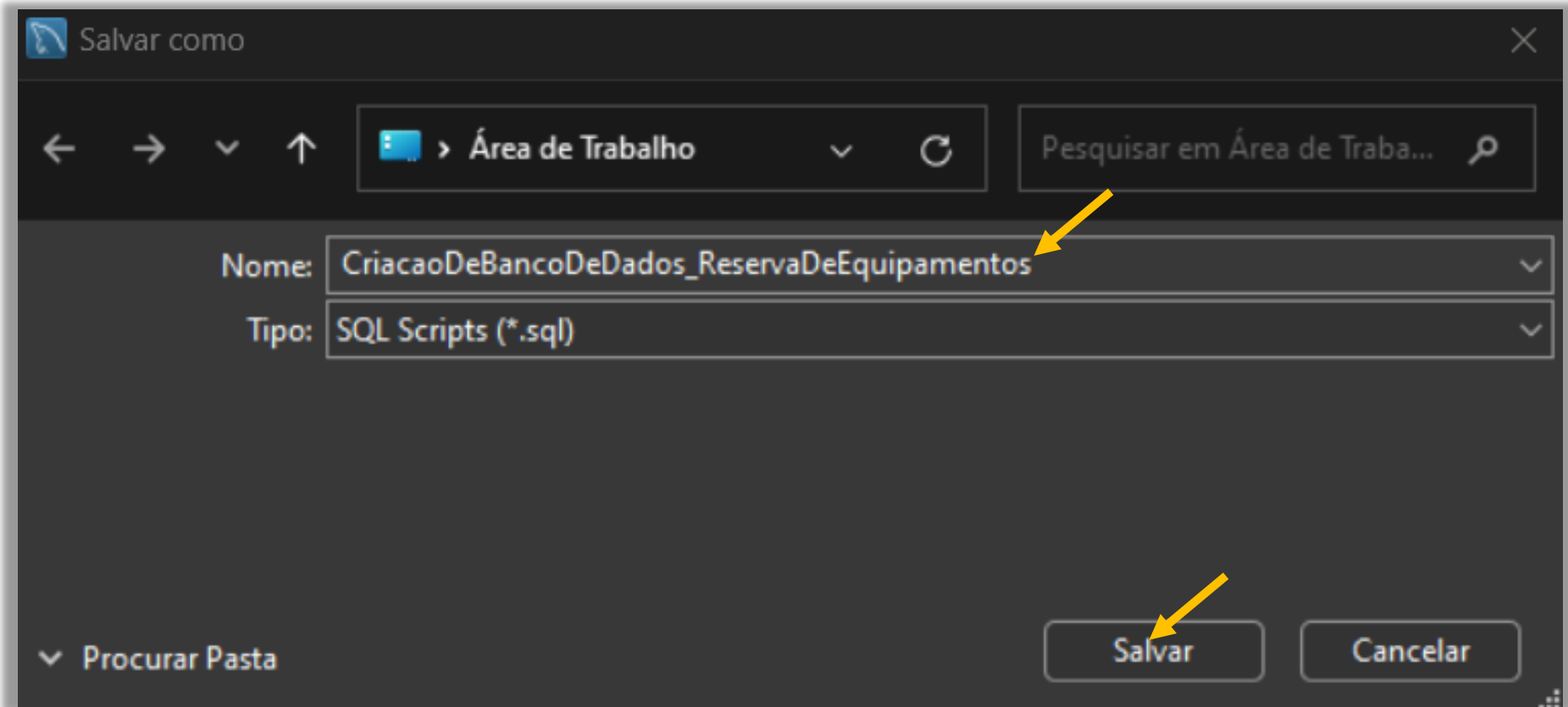
Back

Next

Cancel



# Do Modelo Lógico para o Modelo Físico



# Do Modelo Lógico para o Modelo Físico

```
11  -----
12  -- Schema reserva_de Equipamentos
13  -----
14  CREATE SCHEMA IF NOT EXISTS `reserva_de Equipamentos` DEFAULT CHARACTER SET utf8 ;
15  USE `reserva_de Equipamentos` ;
16
17  -----
18  -- Table `Professor`
19  -----
20  CREATE TABLE IF NOT EXISTS `Professor` (
21    `id` INT NOT NULL AUTO_INCREMENT,
22    `cpf` BIGINT NOT NULL,
23    `nome` VARCHAR(60) NOT NULL,
24    `email` VARCHAR(50) NULL,
```

Save to File...

Copy to Clipboard

Salvar Script SQL Gerado

Back

Next

Cancel



# Do Modelo Lógico para o Modelo Físico

## Forward Engineering Progress

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- ☒ Connect to DBMS
- ☒ Execute Forward Engineered Script
- ☒ Read Back Changes Made by Server
- ☒ Save Synchronization State

Forward Engineer Finished Successfully

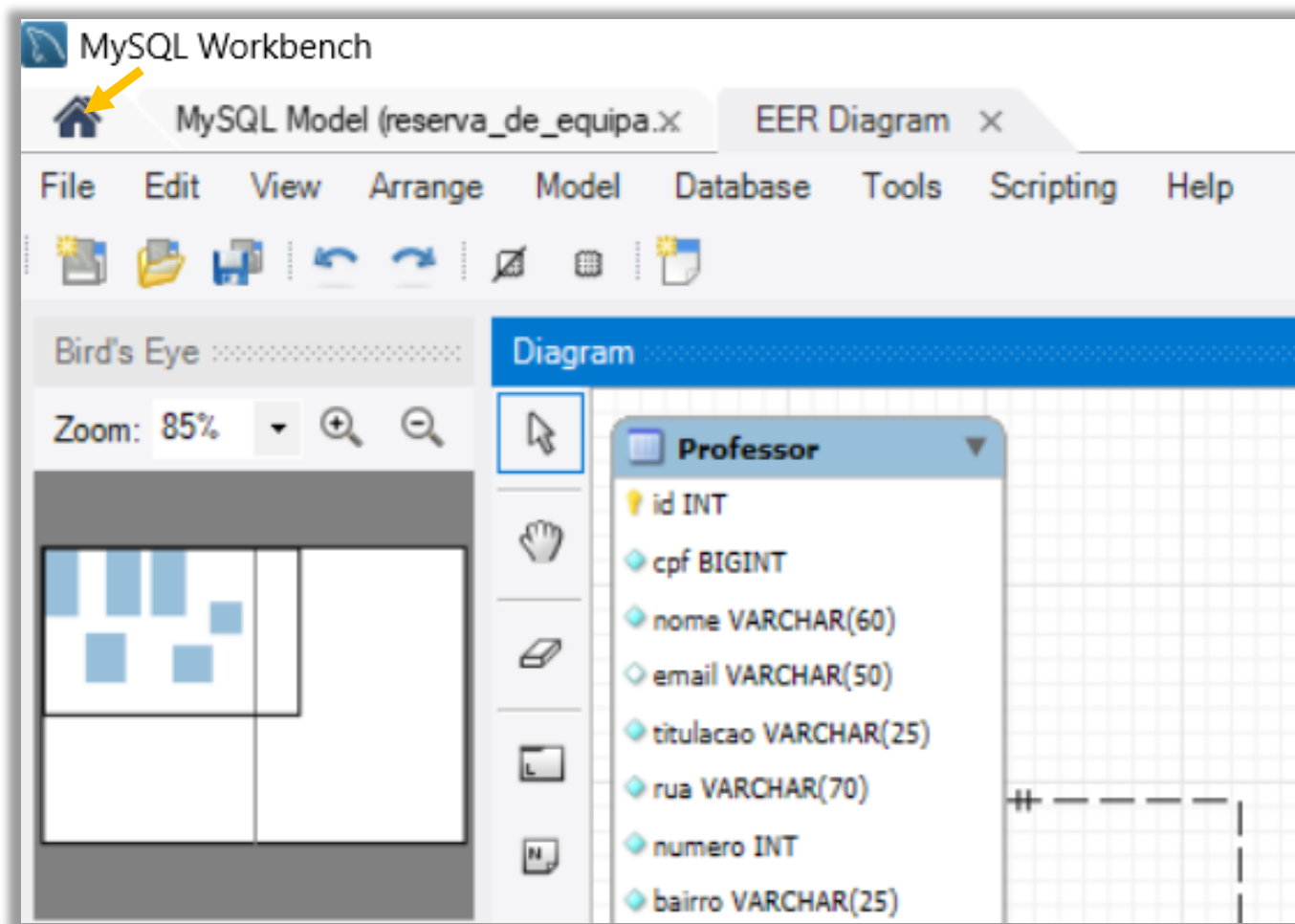
Back

Close

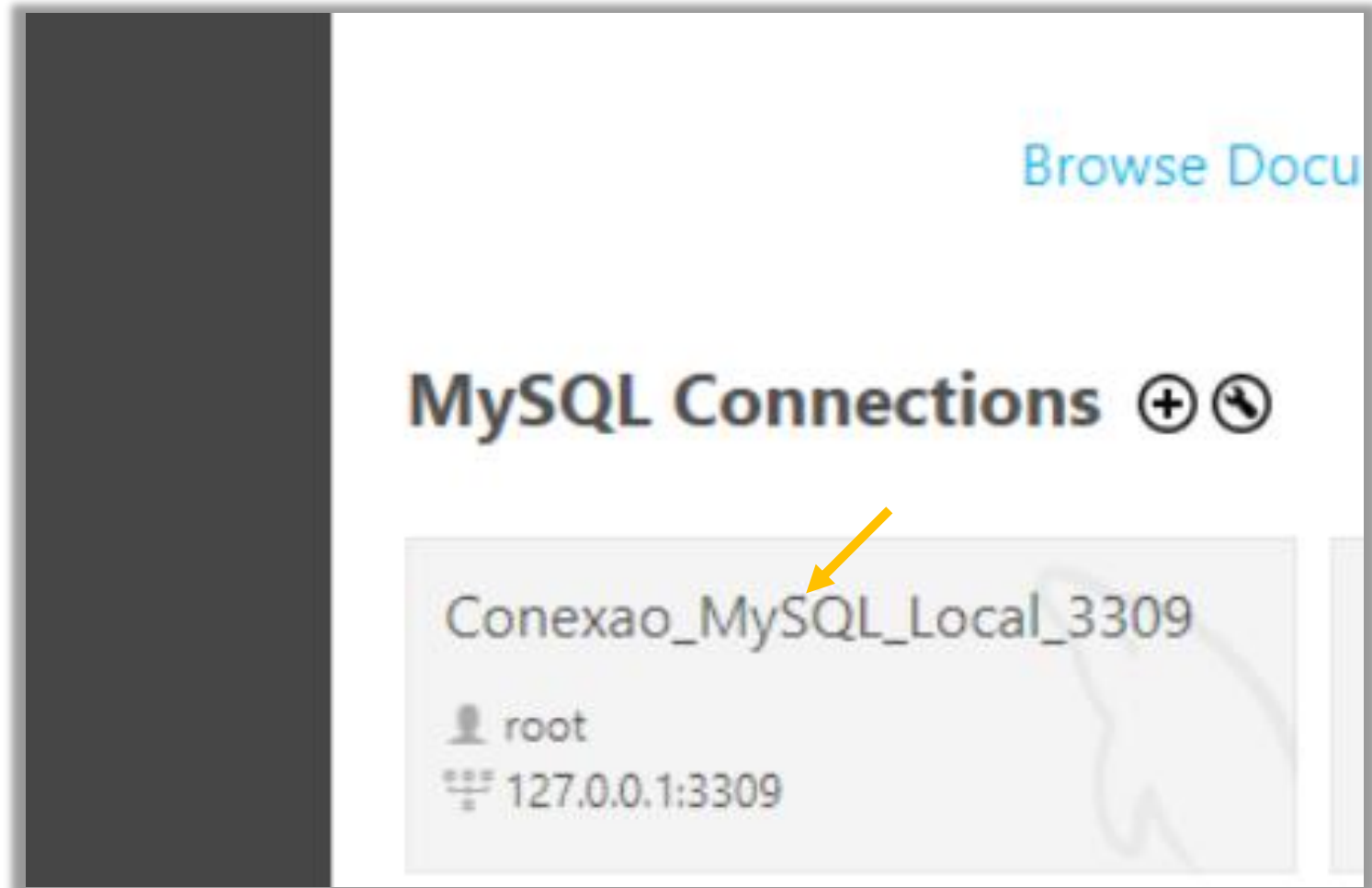
Cancel



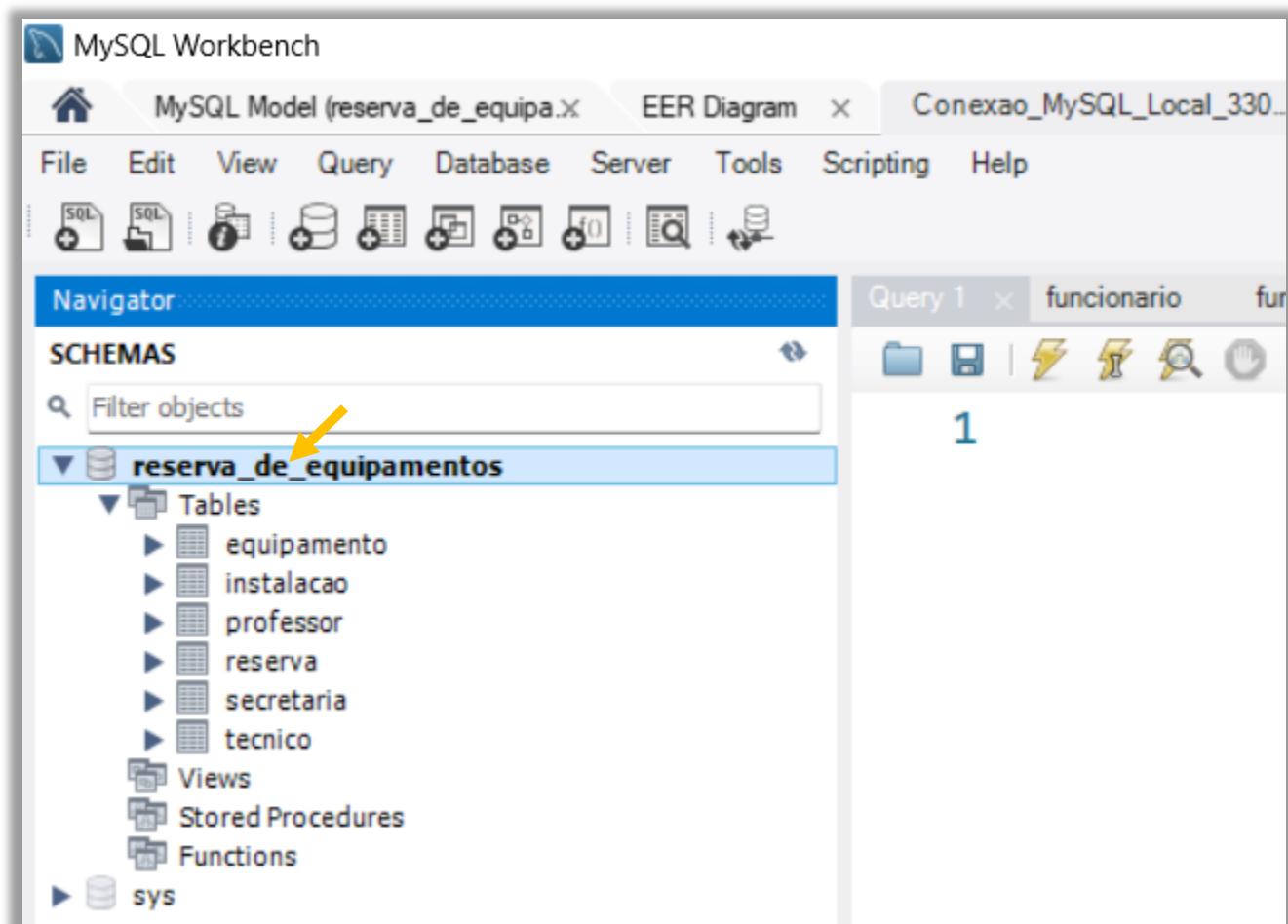
# Do Modelo Lógico para o Modelo Físico



# Do Modelo Lógico para o Modelo Físico

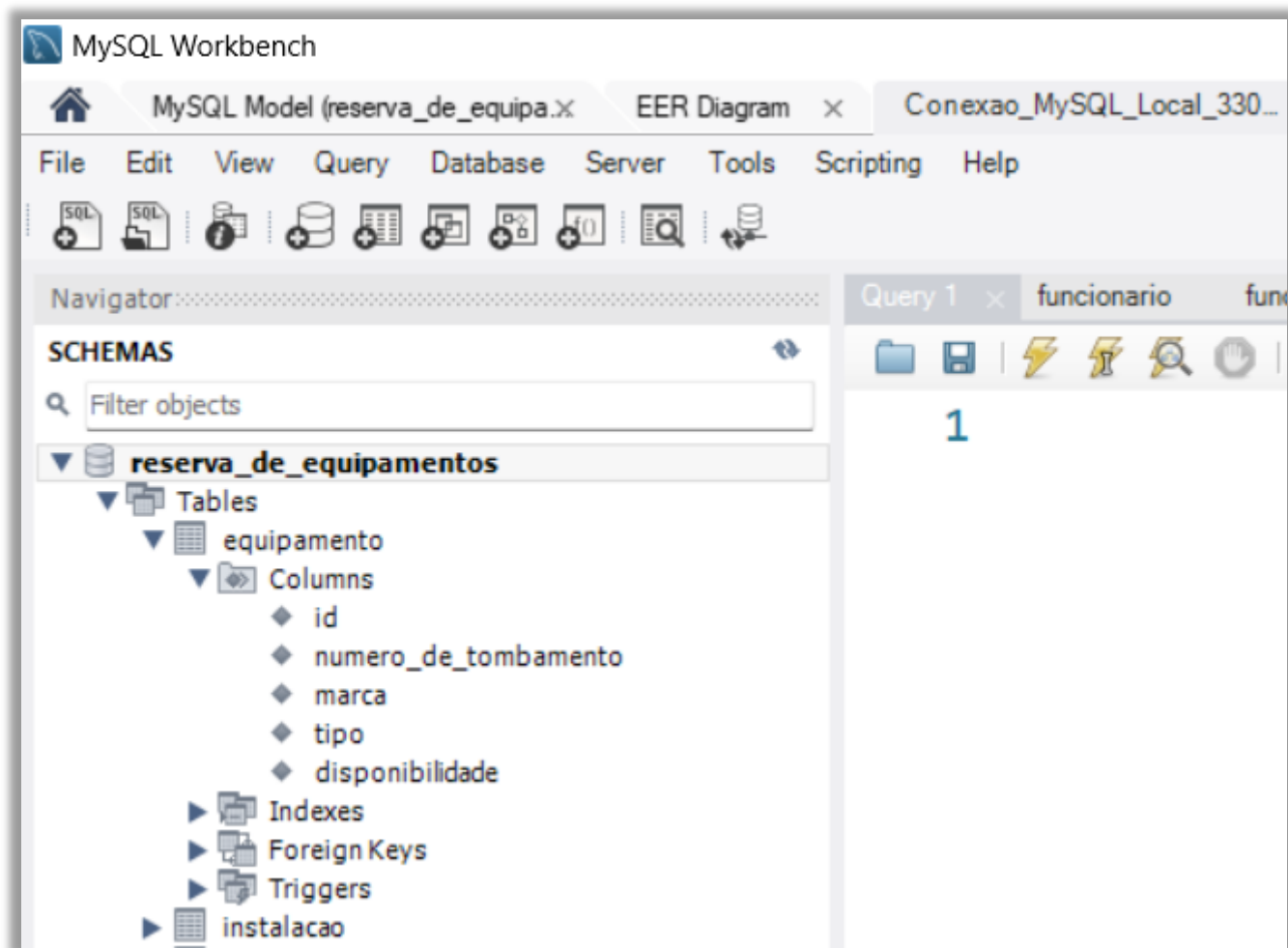


# Do Modelo Lógico para o Modelo Físico

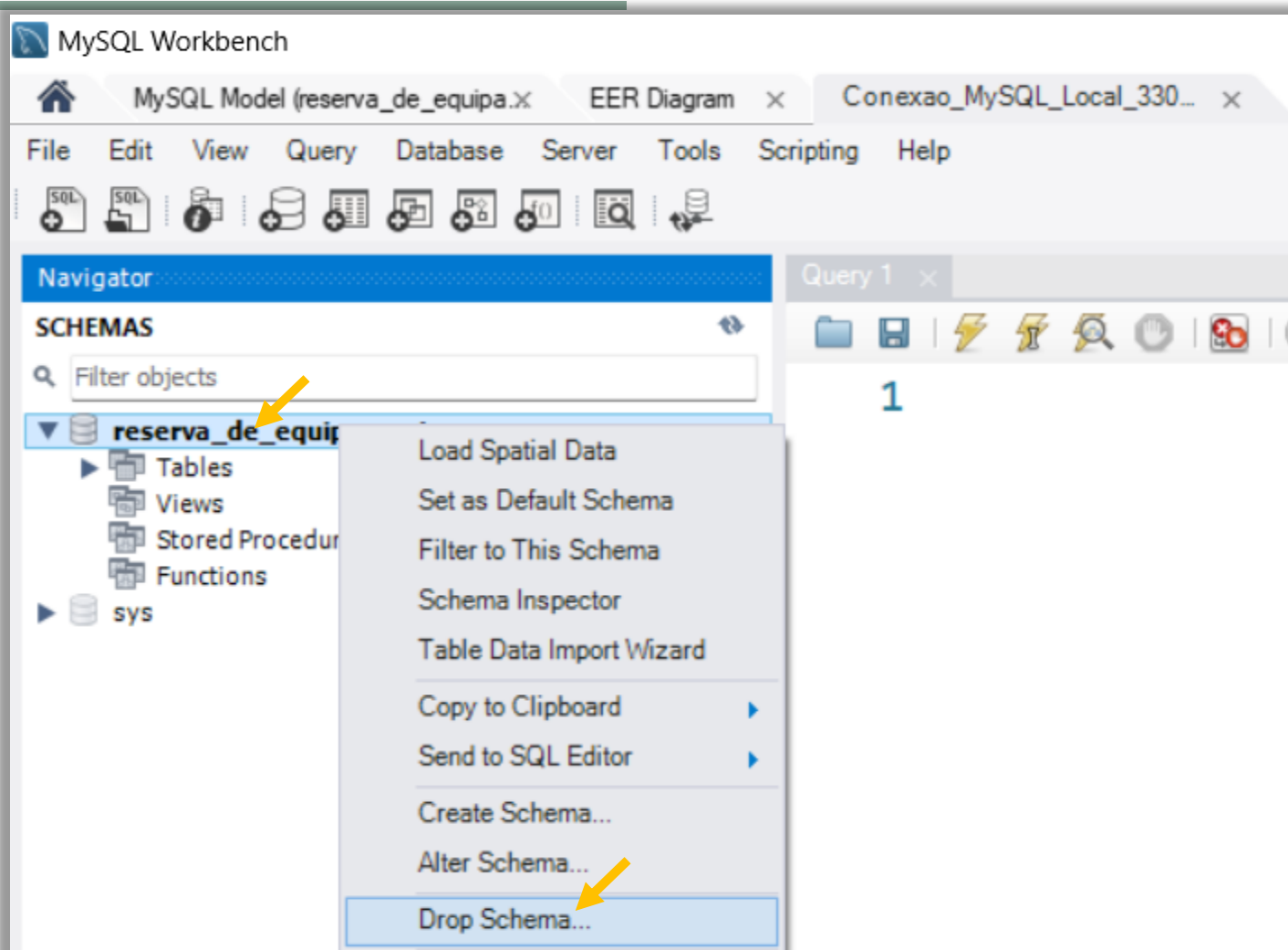




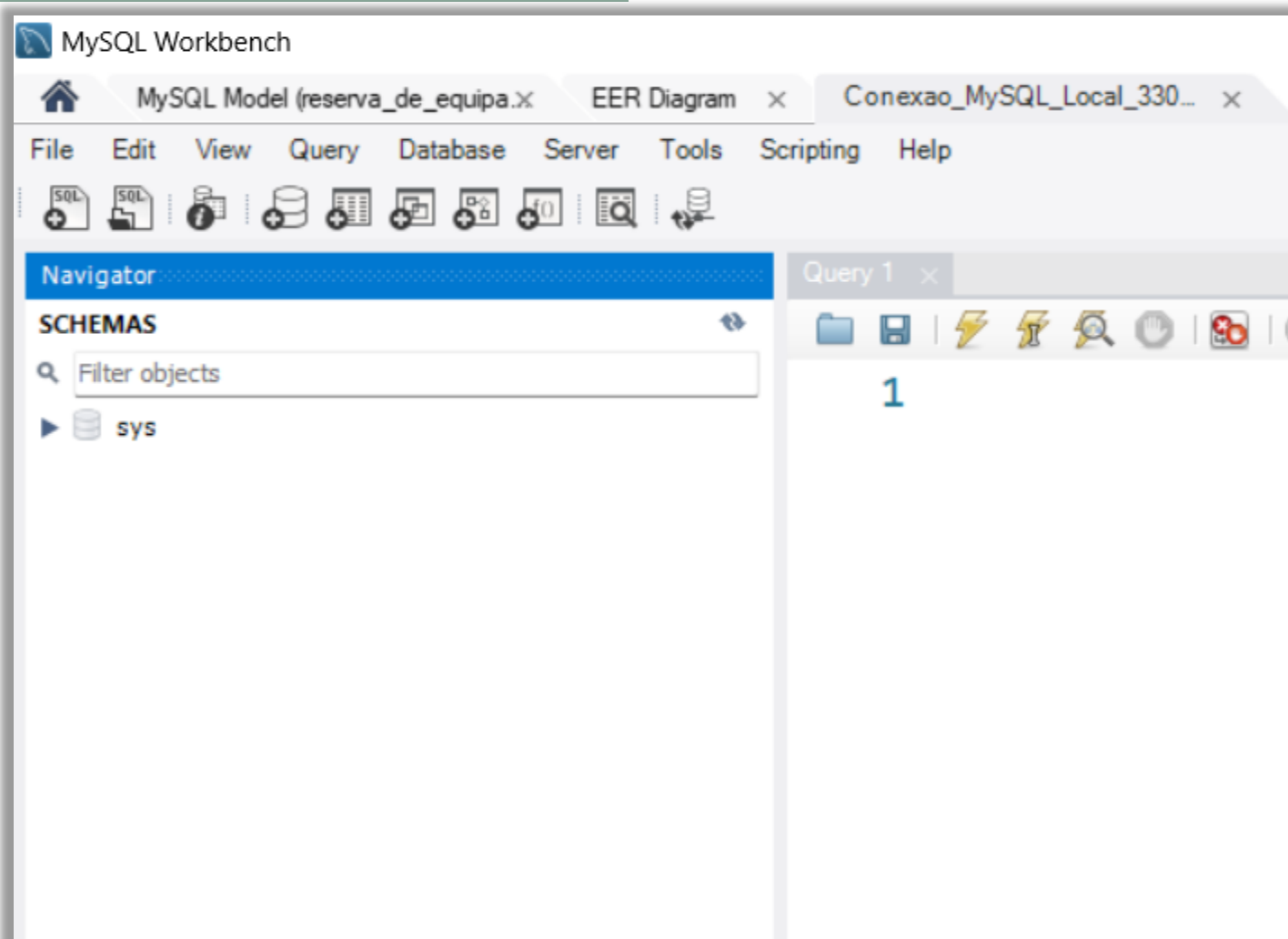
# Do Modelo Lógico para o Modelo Físico



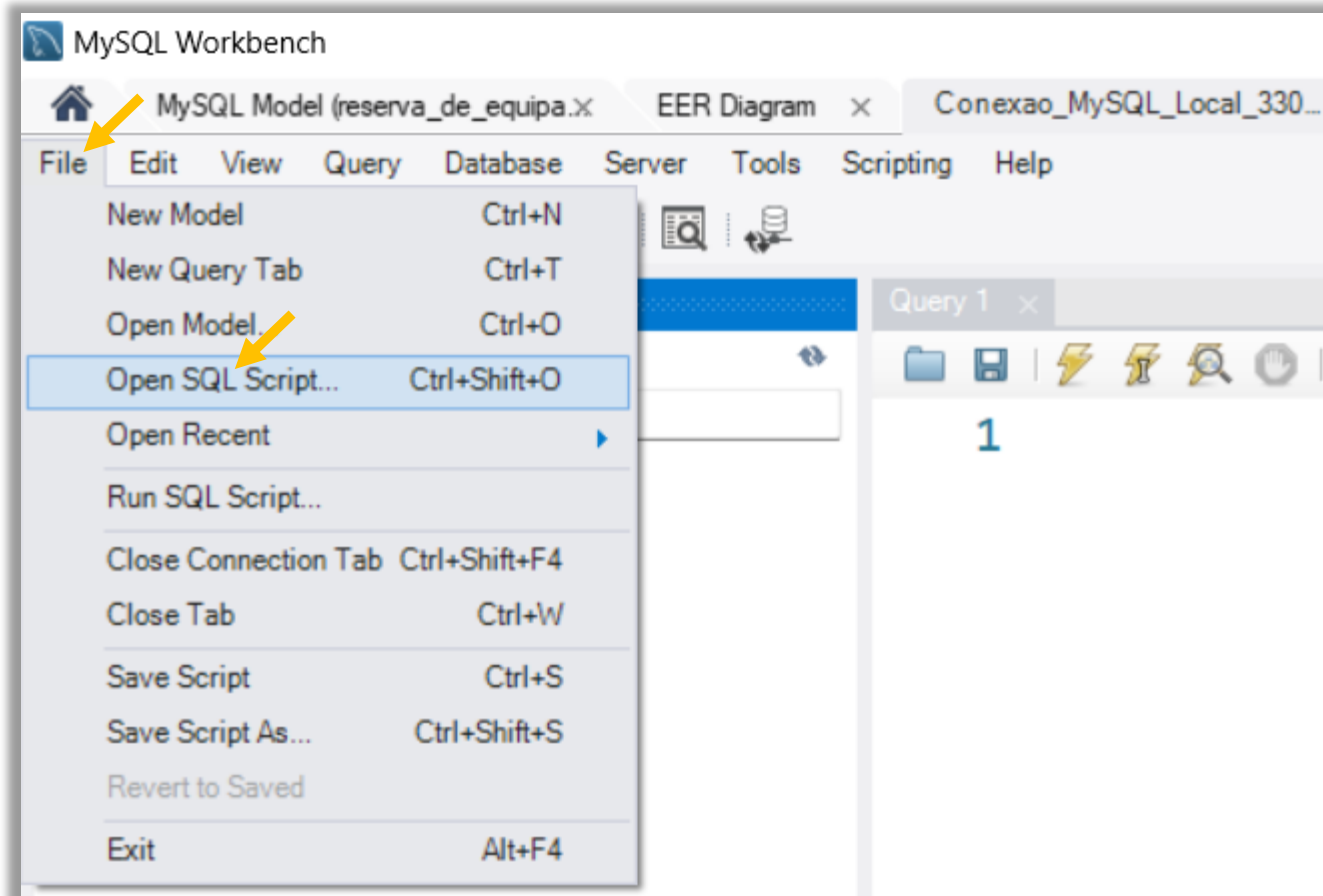
# Executar o Script SQL de criação de DB



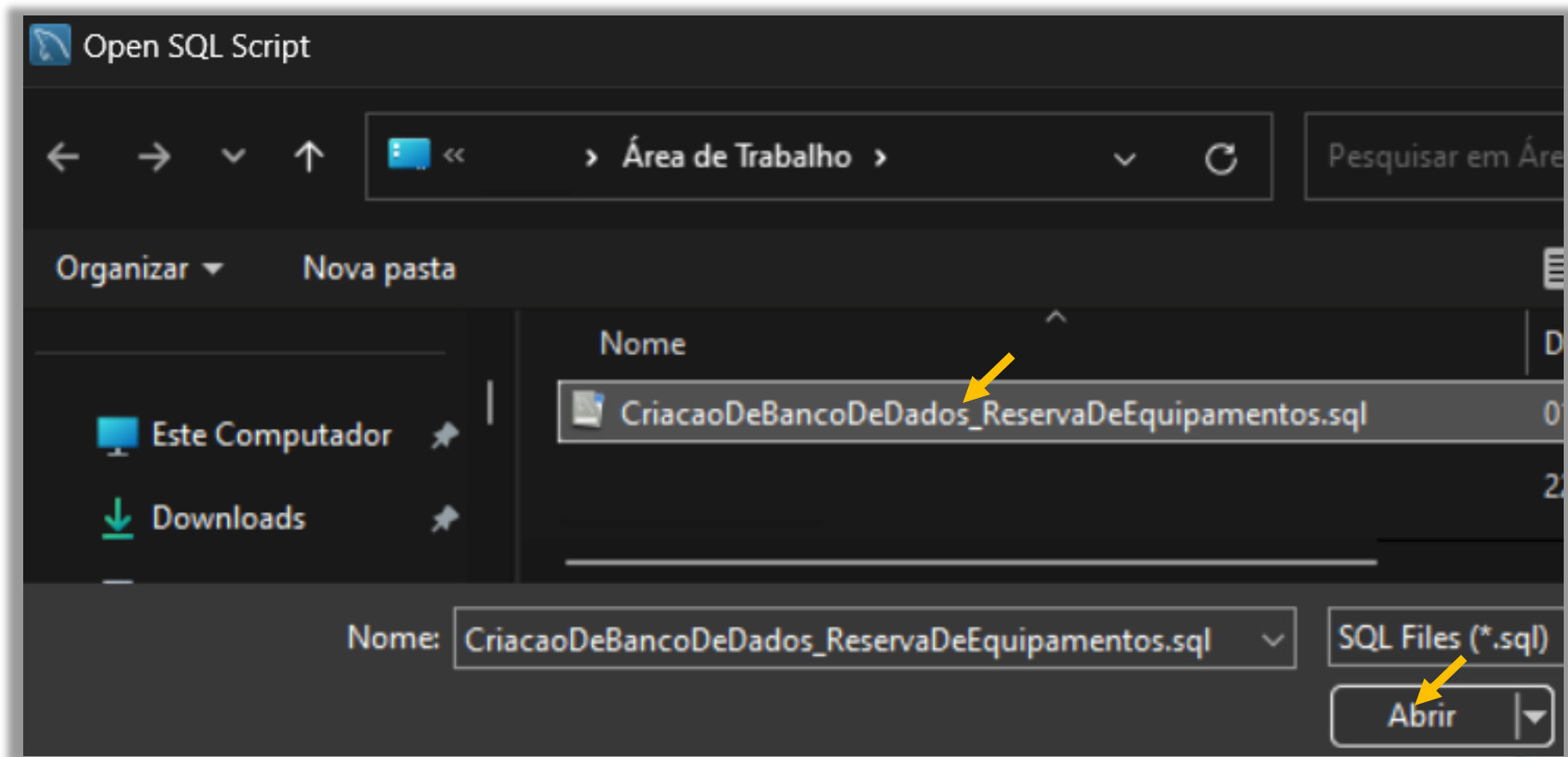
# Executar o Script SQL de criação de DB



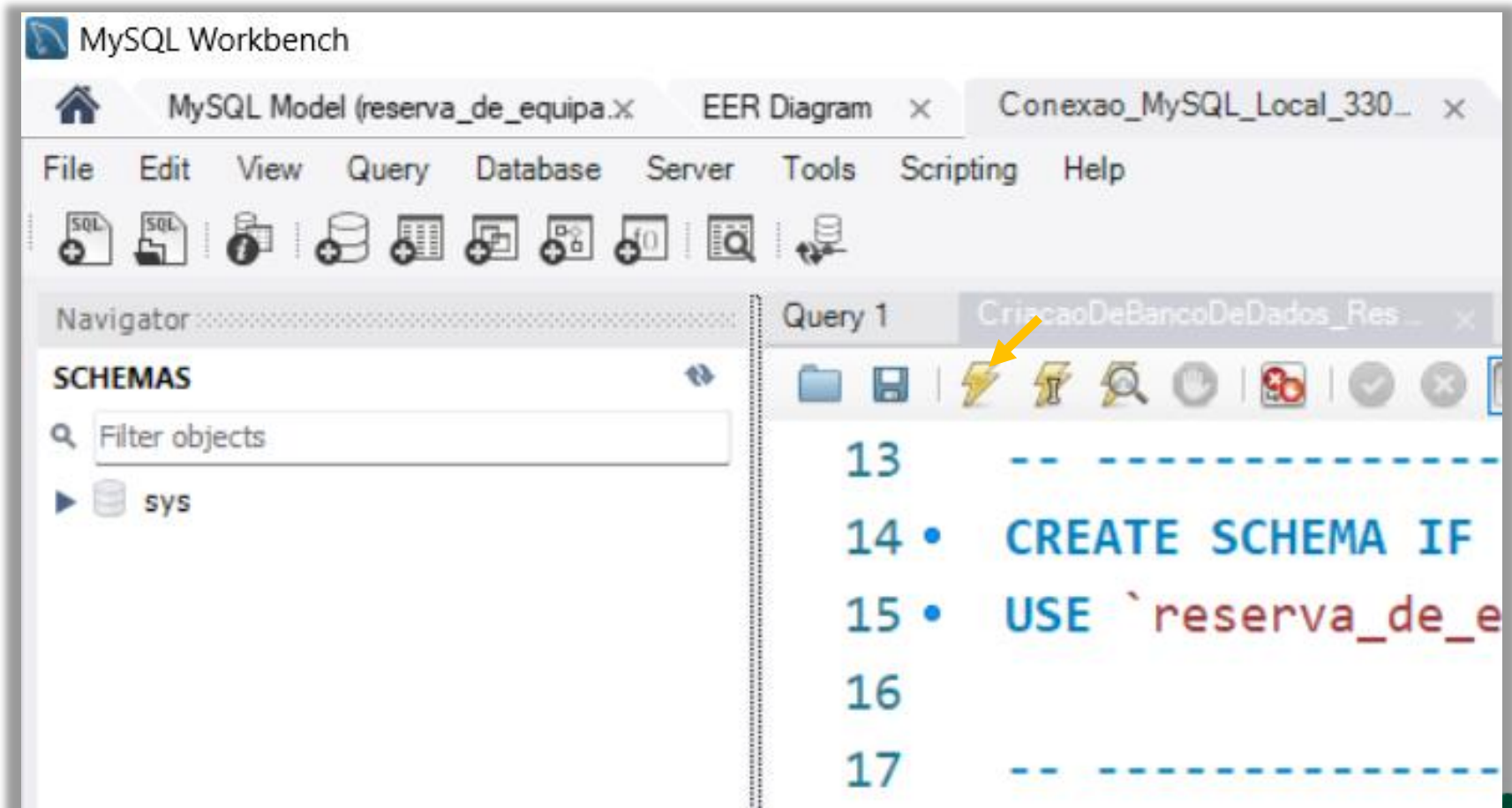
# Executar o Script SQL de criação de DB



# Executar o Script SQL de criação de DB



\_\_\_\_\_



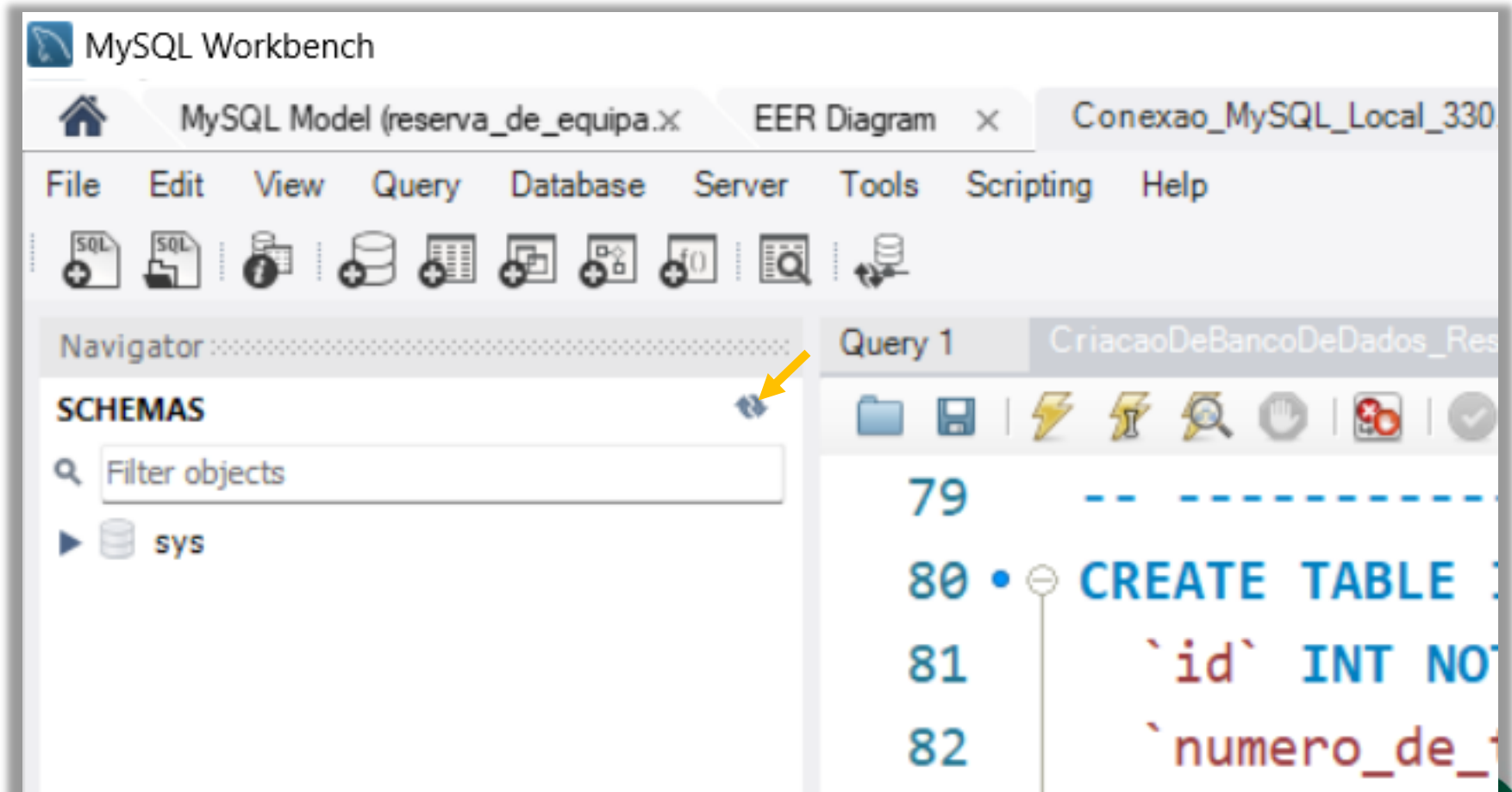
# Executar o Script SQL de criação de DB

Output				
Action Output				
	#	Time	Action	Message
✓	1	17:38:41	DROP DATABASE `reserva_de_equipamentos`	6 row(s) affected
✓	2	17:43:56	SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0	0 row(s) affected
✓	3	17:43:56	SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0	0 row(s) affected
✓	4	17:43:56	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'	0 row(s) affected
⚠	5	17:43:56	CREATE SCHEMA IF NOT EXISTS `reserva_de_equipamentos` DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci	1 row(s) affected,
✓	6	17:43:56	USE `reserva_de_equipamentos`	0 row(s) affected
✓	7	17:43:56	CREATE TABLE IF NOT EXISTS `Professor` ( `id` INT NOT NULL AUTO_INCREMENT, `cpf` VARCHAR(14) NOT NULL, `nome` VARCHAR(45) NOT NULL, `email` VARCHAR(45) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB	0 row(s) affected
✓	8	17:43:56	CREATE TABLE IF NOT EXISTS `Secretaria` ( `id` INT NOT NULL AUTO_INCREMENT, `cpf` VARCHAR(14) NOT NULL, `nome` VARCHAR(45) NOT NULL, `email` VARCHAR(45) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB	0 row(s) affected
✓	9	17:43:56	CREATE TABLE IF NOT EXISTS `Tecnico` ( `id` INT NOT NULL AUTO_INCREMENT, `cpf` VARCHAR(14) NOT NULL, `nome` VARCHAR(45) NOT NULL, `email` VARCHAR(45) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB	0 row(s) affected
⚠	10	17:43:56	CREATE TABLE IF NOT EXISTS `Equipamento` ( `id` INT NOT NULL AUTO_INCREMENT, `nome` VARCHAR(45) NOT NULL, `descricao` VARCHAR(45) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB	0 row(s) affected,
✓	11	17:43:56	CREATE TABLE IF NOT EXISTS `Reserva` ( `id` INT NOT NULL AUTO_INCREMENT, `professor_id` INT NOT NULL, `secretaria_id` INT NOT NULL, `tecnico_id` INT NOT NULL, `equipamento_id` INT NOT NULL, PRIMARY KEY (`id`), INDEX (`professor_id`), INDEX (`secretaria_id`), INDEX (`tecnico_id`), INDEX (`equipamento_id`)) ENGINE=InnoDB	0 row(s) affected
✓	12	17:43:56	CREATE TABLE IF NOT EXISTS `Instalacao` ( `reserva_id` INT NOT NULL, `tecnico_id` INT NOT NULL, PRIMARY KEY (`reserva_id`, `tecnico_id`)) ENGINE=InnoDB	0 row(s) affected
✓	13	17:43:56	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected
✓	14	17:43:56	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected
✓	15	17:43:56	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected



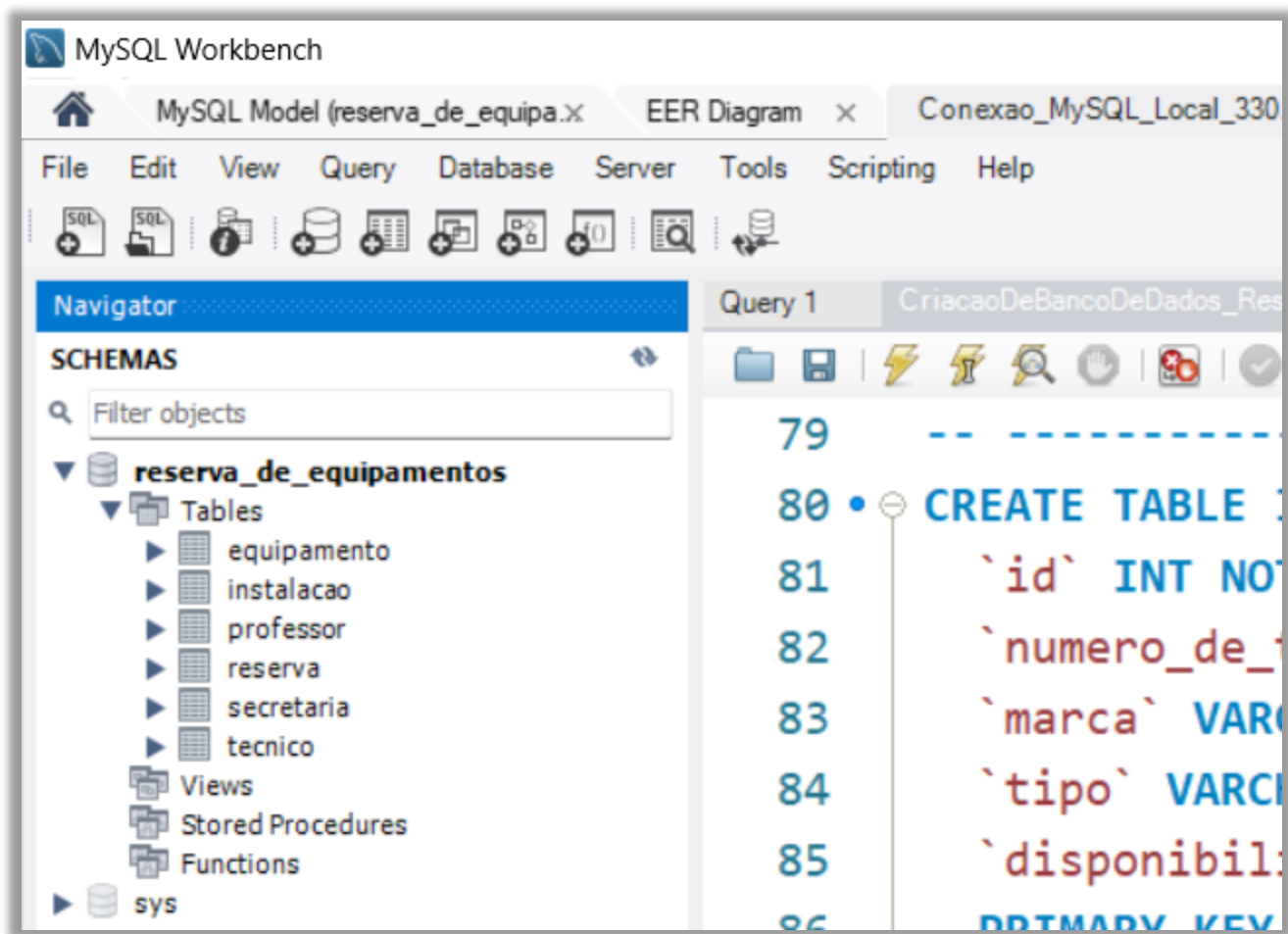


## Executar o Script SQL de criação de DB





# Executar o Script SQL de criação de DB



# Explicação do Script SQL de criação de DB

```
1 • CREATE SCHEMA `reserva_de Equipamentos` ;  
2  
3 • USE `reserva_de Equipamentos` ;
```



# Explicação do Script SQL de criação de DB

```
5 • CREATE TABLE `Professor` (  
6     `id` INT NOT NULL AUTO_INCREMENT,  
7     `cpf` BIGINT NOT NULL,  
8     `nome` VARCHAR(60) NOT NULL,  
9     `email` VARCHAR(50) NULL,  
10    `titulacao` VARCHAR(25) NOT NULL,  
11    `rua` VARCHAR(70) NOT NULL,  
12    `numero` INT NOT NULL,  
13    `bairro` VARCHAR(25) NOT NULL,  
14    `cidade` VARCHAR(45) NOT NULL,  
15    `cep` BIGINT NOT NULL,  
16    `estado` VARCHAR(40) NOT NULL,  
17    PRIMARY KEY (`id`),  
18    UNIQUE INDEX `cpf_UNIQUE` (`cpf`));
```



# Explicação do Script SQL de criação de DB

```
20 • CREATE TABLE `Secretaria` (  
21     `id` INT NOT NULL AUTO_INCREMENT,  
22     `cpf` BIGINT NOT NULL,  
23     `nome` VARCHAR(60) NOT NULL,  
24     `email` VARCHAR(50) NOT NULL,  
25     `turno` VARCHAR(15) NOT NULL,  
26     `rua` VARCHAR(70) NOT NULL,  
27     `numero` INT NOT NULL,  
28     `bairro` VARCHAR(25) NOT NULL,  
29     `cidade` VARCHAR(45) NOT NULL,  
30     `cep` BIGINT NOT NULL,  
31     `estado` VARCHAR(40) NOT NULL,  
32     PRIMARY KEY (`id`),  
33     UNIQUE INDEX `cpf_UNIQUE` (`cpf`));
```





# Explicação do Script SQL de criação de DB

```
35 • CREATE TABLE `Tecnico` (  
36     `id` INT NOT NULL AUTO_INCREMENT,  
37     `cpf` BIGINT NOT NULL,  
38     `nome` VARCHAR(60) NOT NULL,  
39     `email` VARCHAR(50) NOT NULL,  
40     `funcao` VARCHAR(25) NOT NULL,  
41     `rua` VARCHAR(70) NOT NULL,  
42     `numero` INT NOT NULL,  
43     `bairro` VARCHAR(25) NOT NULL,  
44     `cidade` VARCHAR(45) NOT NULL,  
45     `cep` BIGINT NOT NULL,  
46     `estado` VARCHAR(40) NOT NULL,  
47     PRIMARY KEY (`id`),  
48     UNIQUE INDEX `cpf_UNIQUE` (`cpf`));
```



# Explicação do Script SQL de criação de DB

```
50 • CREATE TABLE `Equipamento` (  
51     `id` INT NOT NULL AUTO_INCREMENT,  
52     `numero_de_tombamento` INT NOT NULL,  
53     `marca` VARCHAR(25) NOT NULL,  
54     `tipo` VARCHAR(20) NOT NULL,  
55     `disponibilidade` TINYINT(1) NOT NULL,  
56     PRIMARY KEY (`id`),  
57     UNIQUE INDEX `numero_de_tombamento_UNIQUE`  
58     (`numero_de_tombamento`));
```



```
60 • CREATE TABLE `Reserva` (  
61     `id` INT NOT NULL AUTO_INCREMENT,  
62     `professor_id` INT NOT NULL,  
63     `secretaria_id` INT NOT NULL,  
64     `local_de_instalacao` VARCHAR(30) NOT NULL,  
65     `data_da_aula` DATE NOT NULL,  
66     `hora_da_aula` TIME NOT NULL,  
67     `data_de_solicitacao` DATE NOT NULL,  
68     PRIMARY KEY (`id`),  
69     CONSTRAINT `FK_Reserva_Professor`  
70         FOREIGN KEY (`professor_id`)  
71         REFERENCES `Professor` (`id`)  
72         ON DELETE NO ACTION  
73         ON UPDATE NO ACTION,  
74     CONSTRAINT `FK_Reserva_Secretaria`  
75         FOREIGN KEY (`secretaria_id`)  
76         REFERENCES `Secretaria` (`id`)  
77         ON DELETE NO ACTION  
78         ON UPDATE NO ACTION);
```



```
80 • ○ CREATE TABLE `Instalacao` (  
81     `reserva_id` INT NOT NULL,  
82     `tecnico_id` INT NOT NULL,  
83     `equipamento_id` INT NOT NULL,  
84     `data` DATE NOT NULL,  
85     PRIMARY KEY (`reserva_id`),  
86     CONSTRAINT `FK_Instalacao_Tecnico`  
87         FOREIGN KEY (`tecnico_id`)  
88         REFERENCES `Tecnico` (`id`)  
89         ON DELETE NO ACTION  
90         ON UPDATE NO ACTION,  
91     CONSTRAINT `FK_Instalacao_Equipamento`  
92         FOREIGN KEY (`equipamento_id`)  
93         REFERENCES `Equipamento` (`id`)  
94         ON DELETE NO ACTION  
95         ON UPDATE NO ACTION,  
96     CONSTRAINT `FK_Instalacao_Reserva`  
97         FOREIGN KEY (`reserva_id`)  
98         REFERENCES `Reserva` (`id`)  
99         ON DELETE NO ACTION  
100        ON UPDATE NO ACTION);
```



# Exercícios

---

## Exercícios:

1. Usando o Workbench execute todos os exemplos apresentados nos slides anteriores.



# Obrigado!

# Questões?

