



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Bahia

Banco de Dados

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



Roteiro

- Gerando o Modelo Relacional (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.



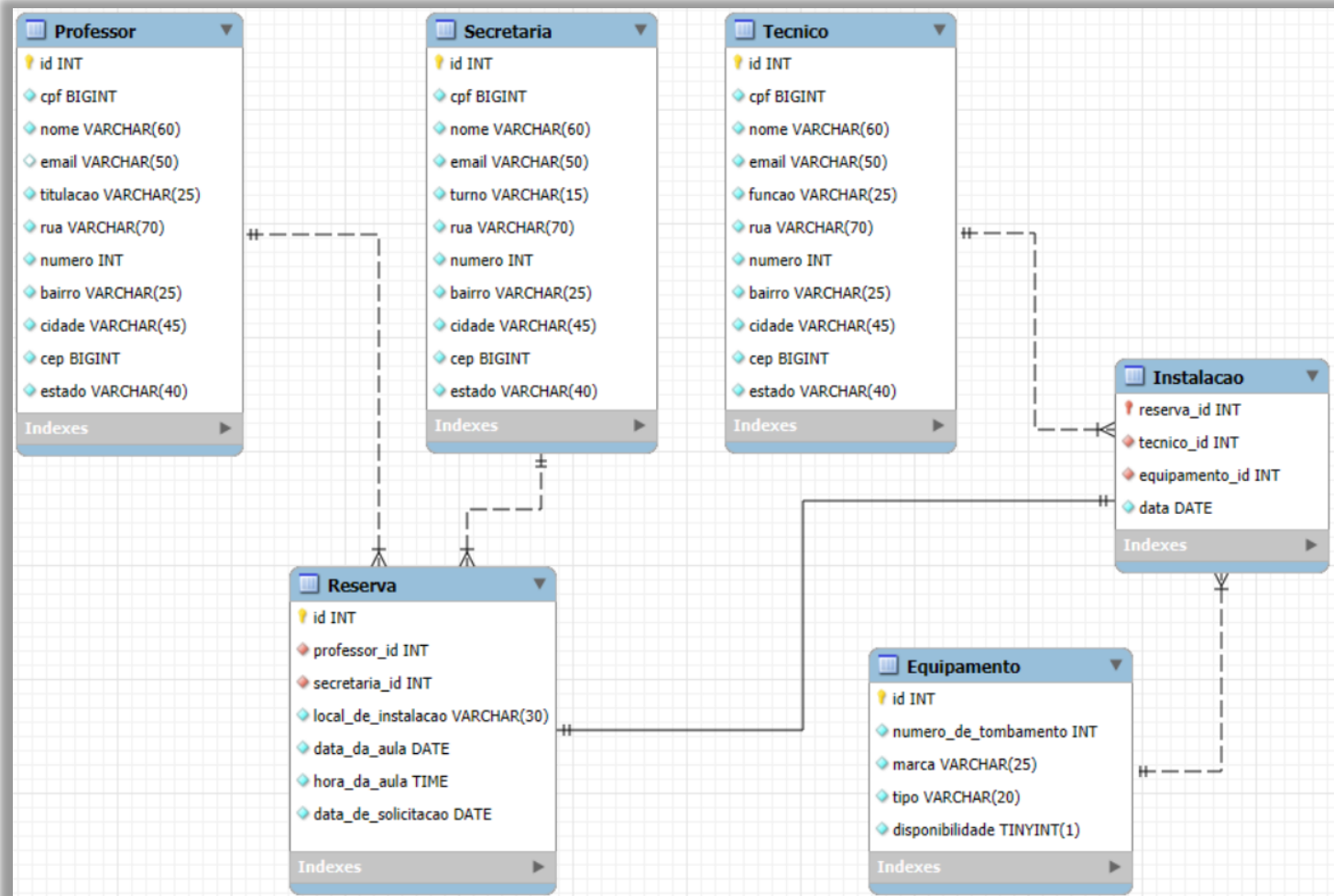
Componentes do Modelo Físico

Modelo Relacional Físico:

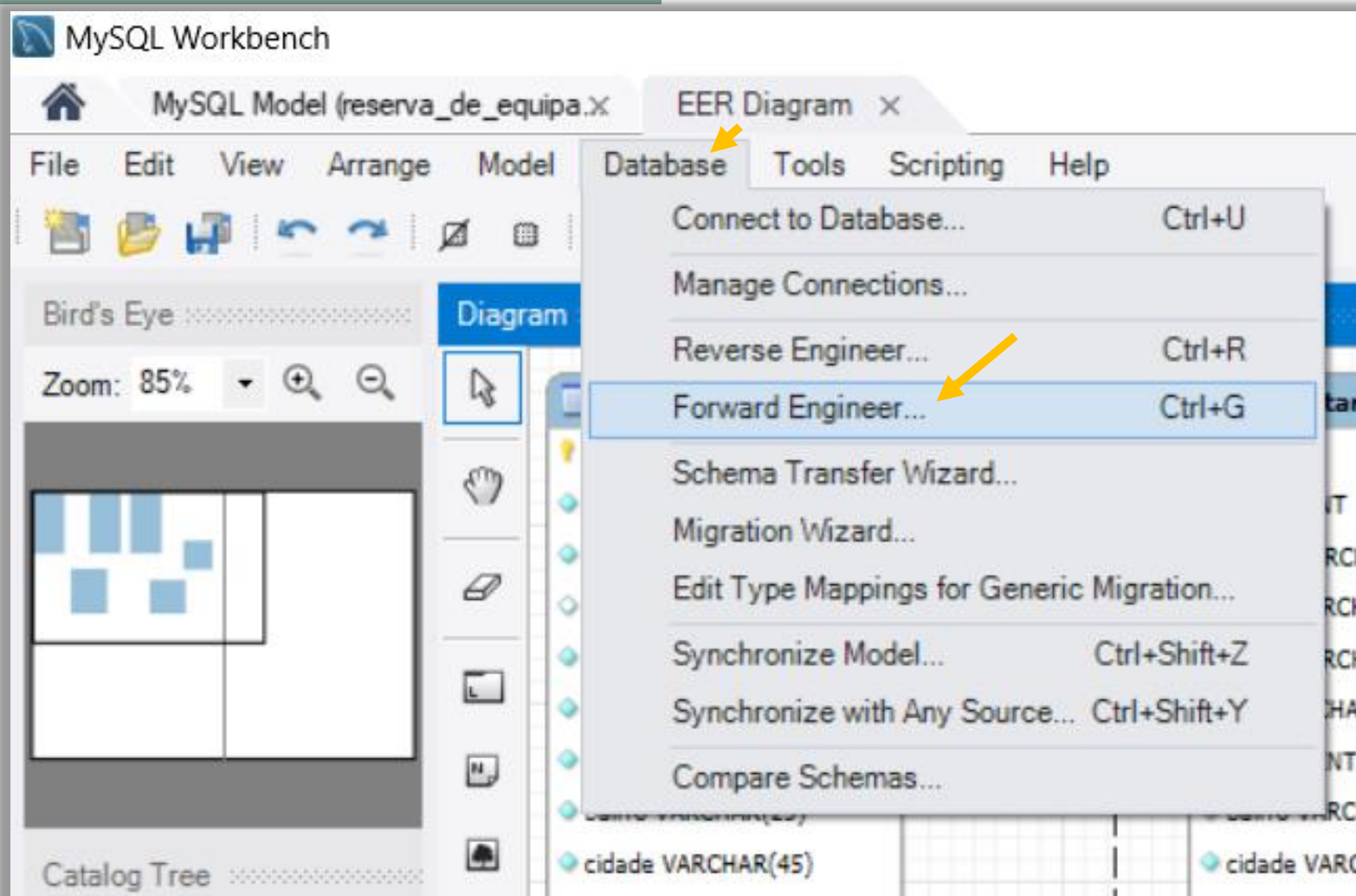
- Estrutura de armazenamento em disco;
- Índices;
- Particionamento de tabelas;
- Estratégias de otimização de desempenho.



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

	<input checked="" type="checkbox"/> Export MySQL Table Objects	Show Filter
6 Total Objects, 6 Selected		
	<input type="checkbox"/> Export MySQL View Objects	Show Filter
0 Total Objects, 0 Selected		
	<input type="checkbox"/> Export MySQL Routine Objects	Show Filter
0 Total Objects, 0 Selected		
	<input type="checkbox"/> Export MySQL Trigger Objects	Show Filter
0 Total Objects, 0 Selected		
	<input type="checkbox"/> Export User Objects	Show Filter
0 Total Objects, 0 Selected		



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

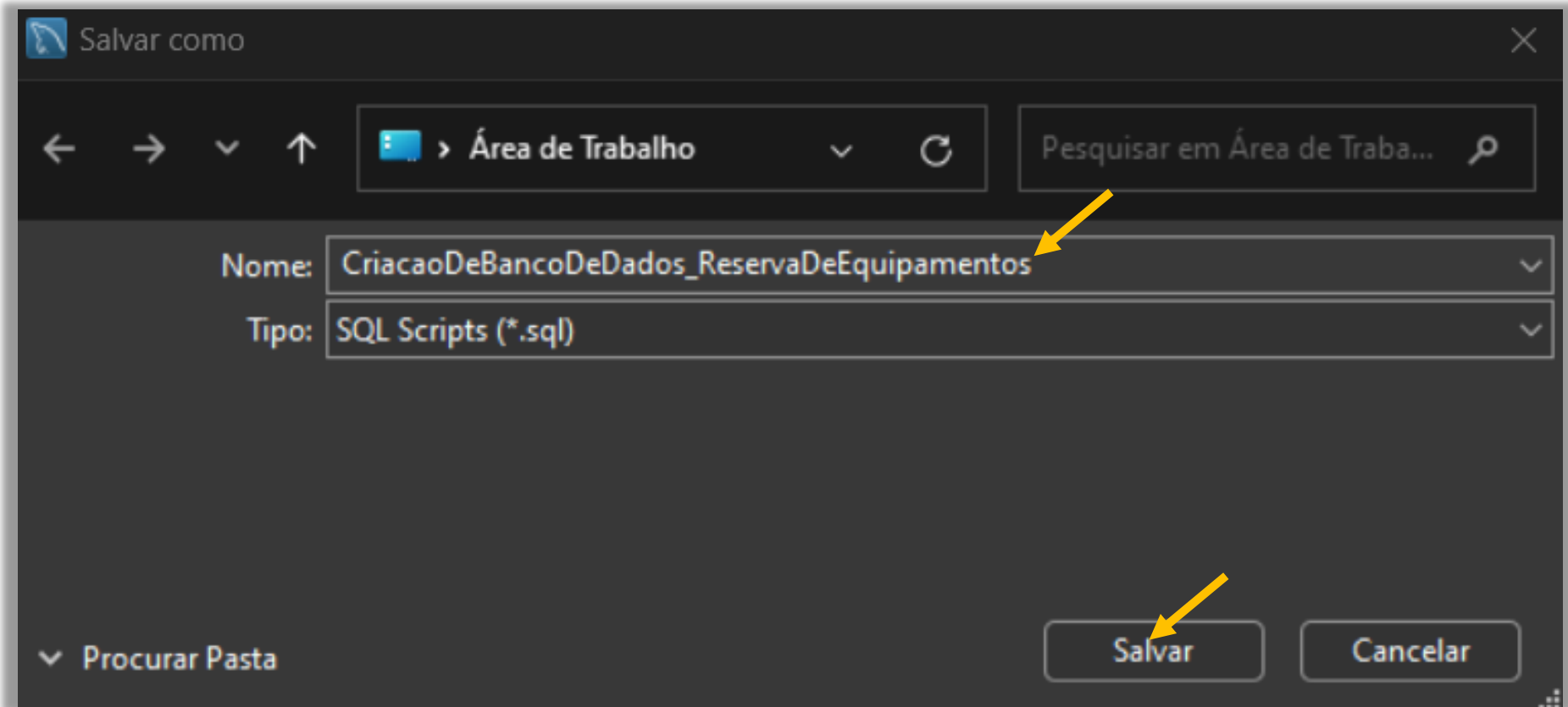
Back

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

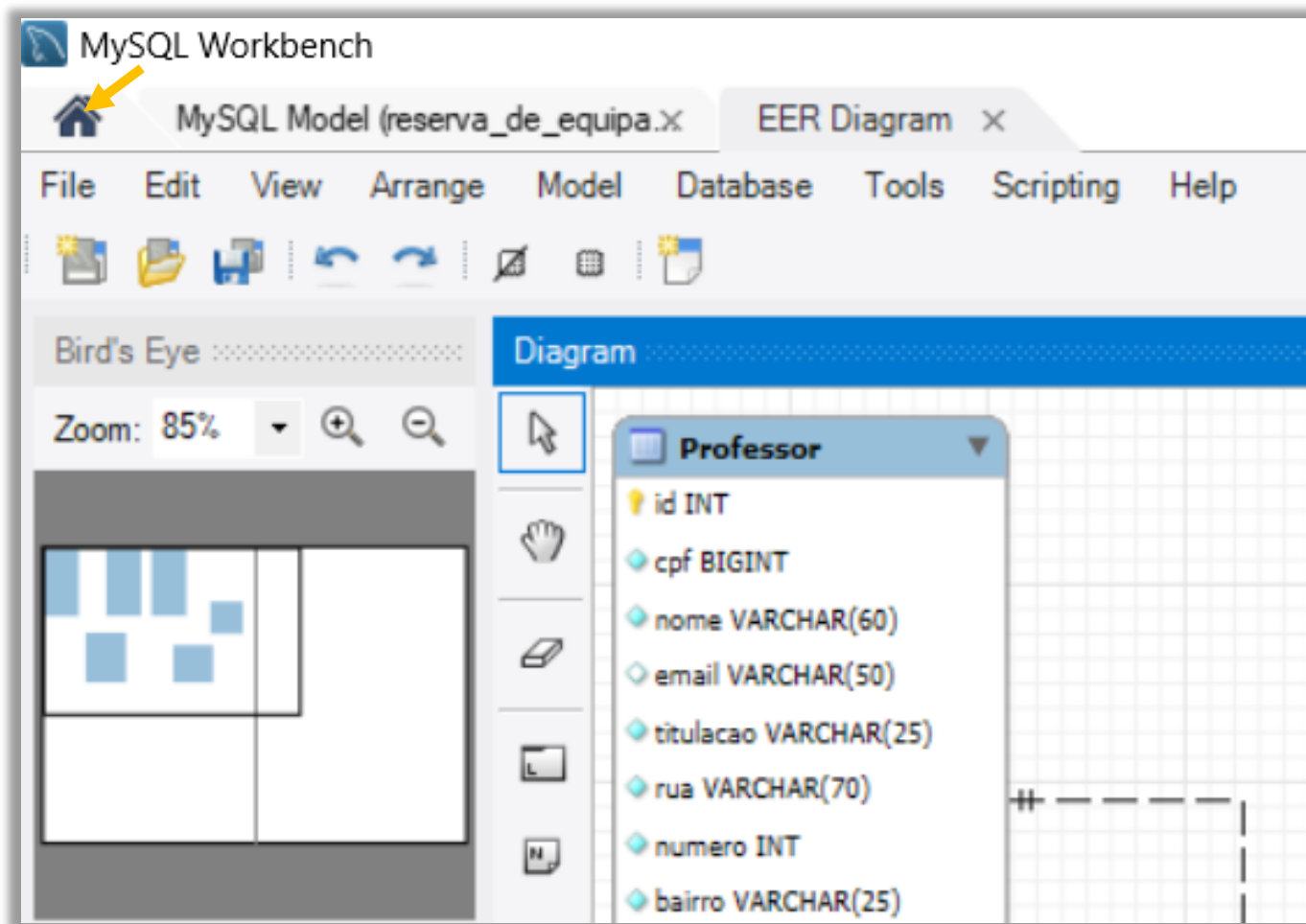
Back

Close

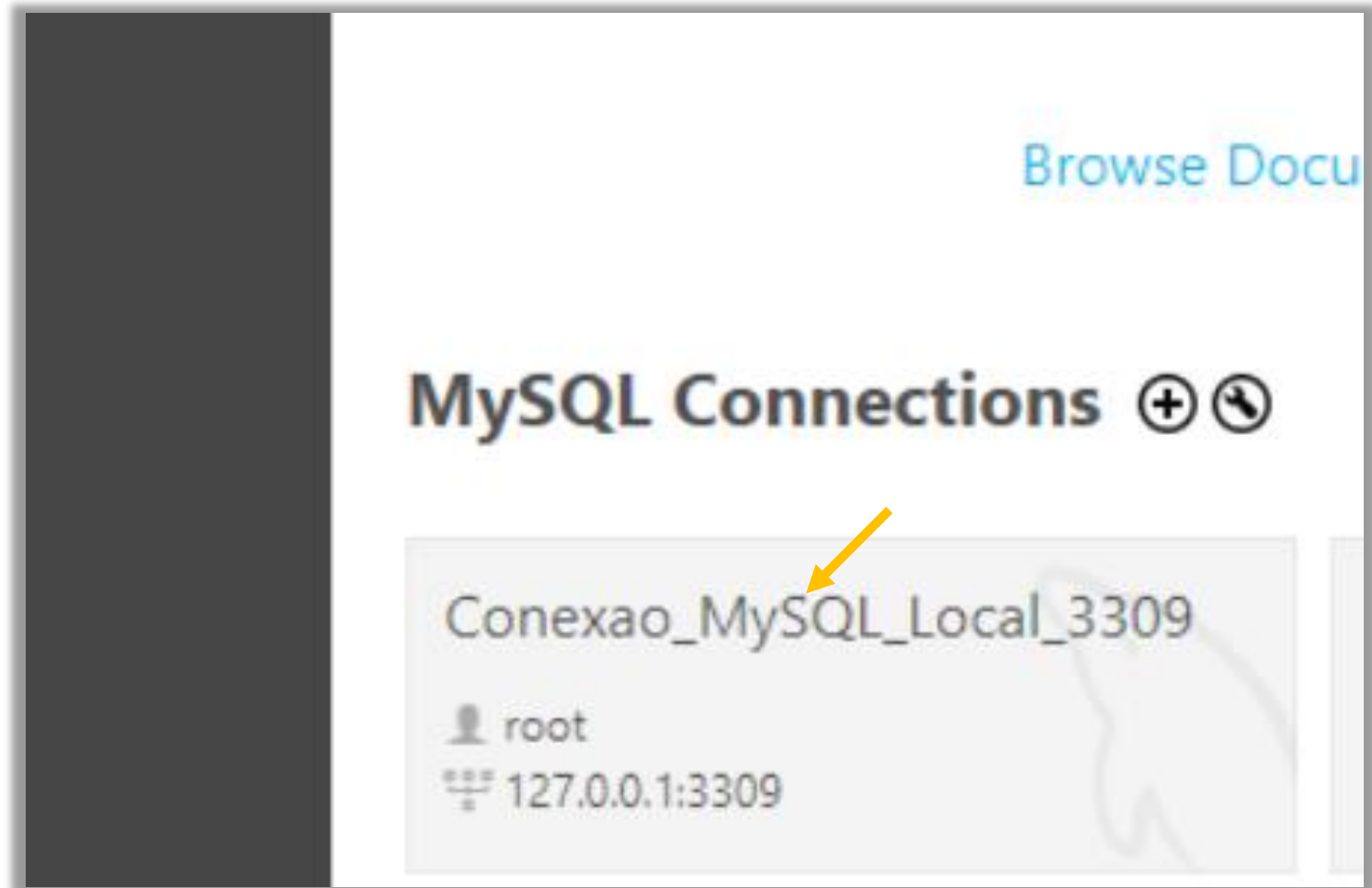
Cancel



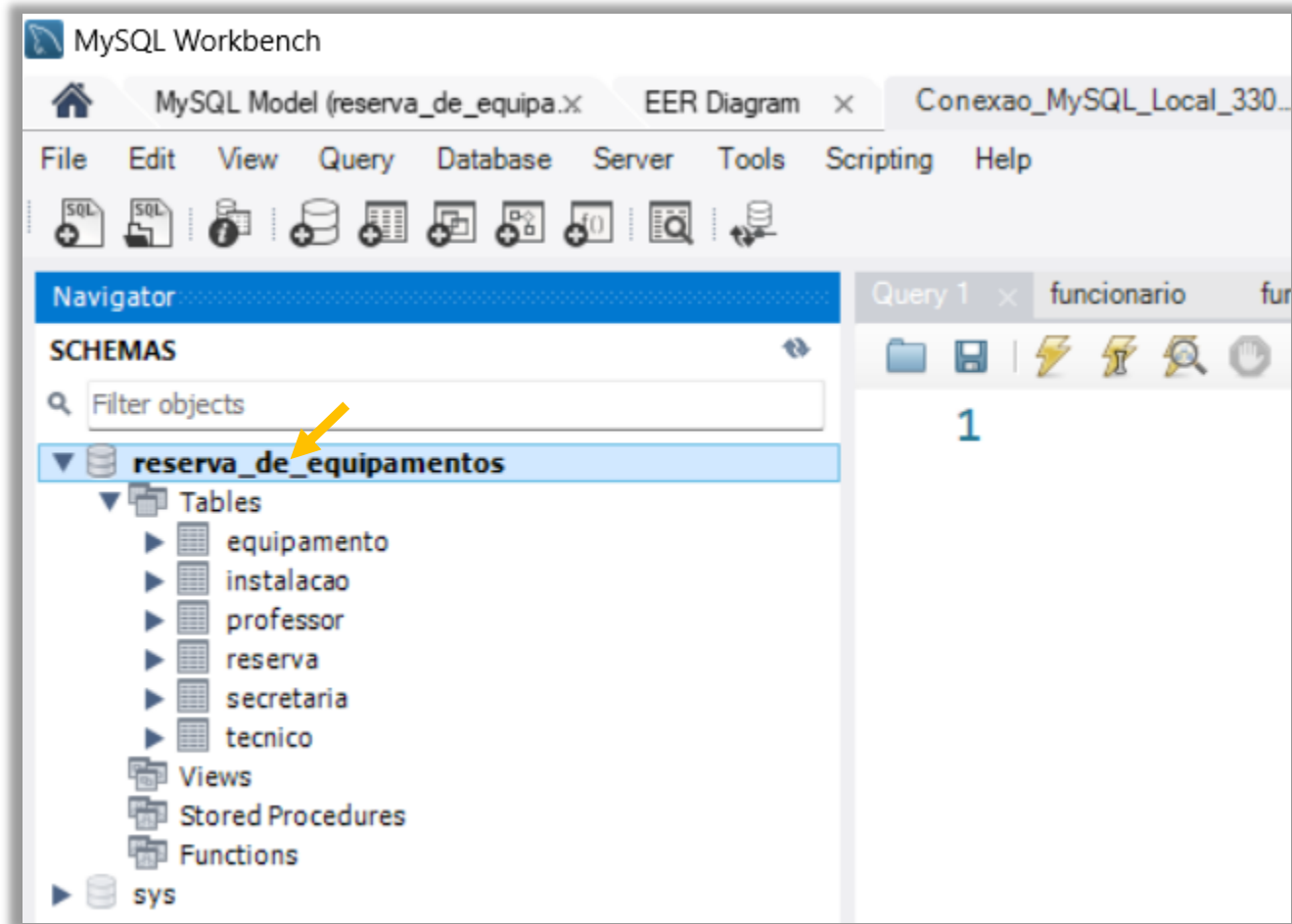
Do Modelo Lógico para o Modelo Físico



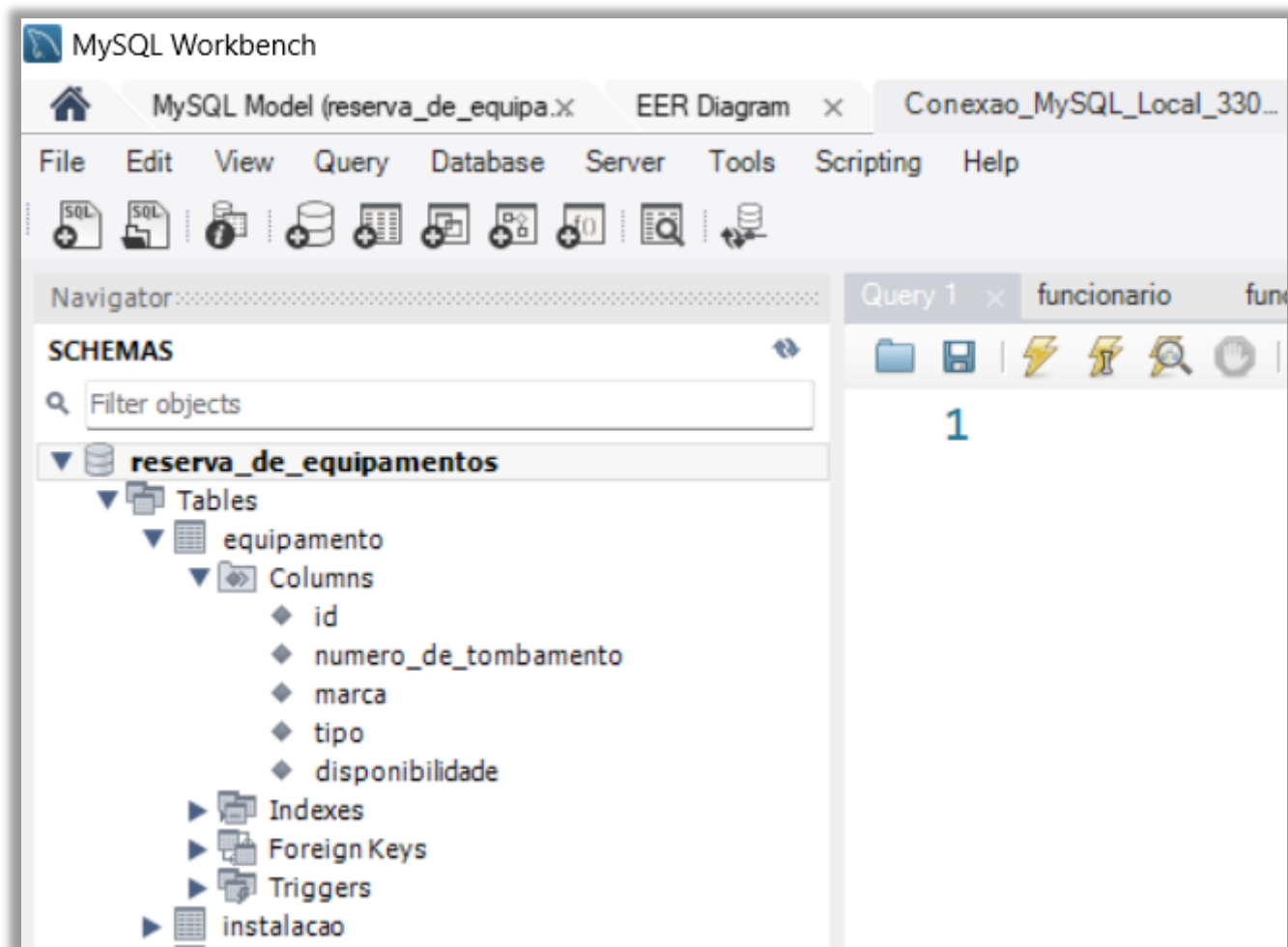
Do Modelo Lógico para o Modelo Físico



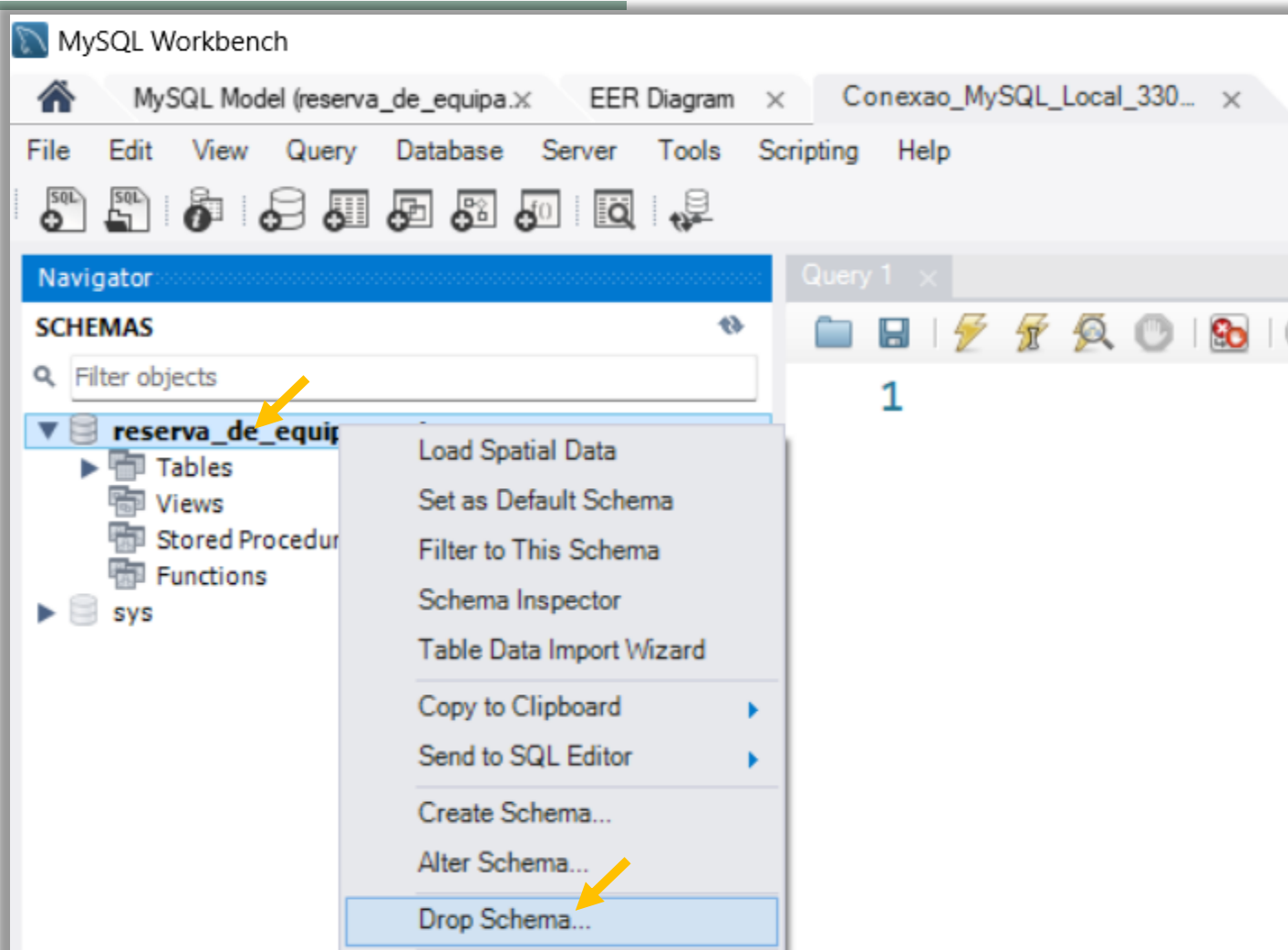
Do Modelo Lógico para o Modelo Físico



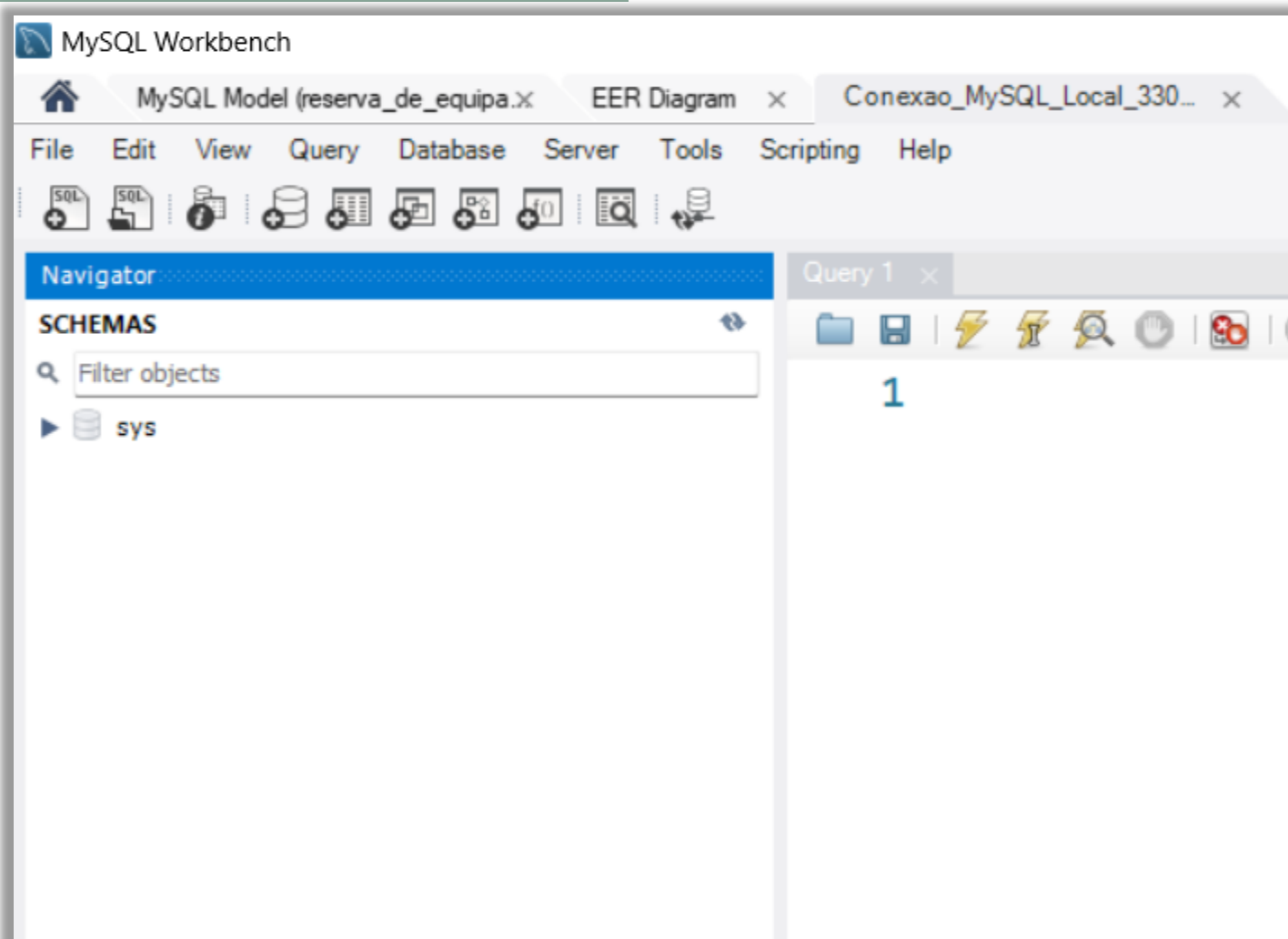
Do Modelo Lógico para o Modelo Físico



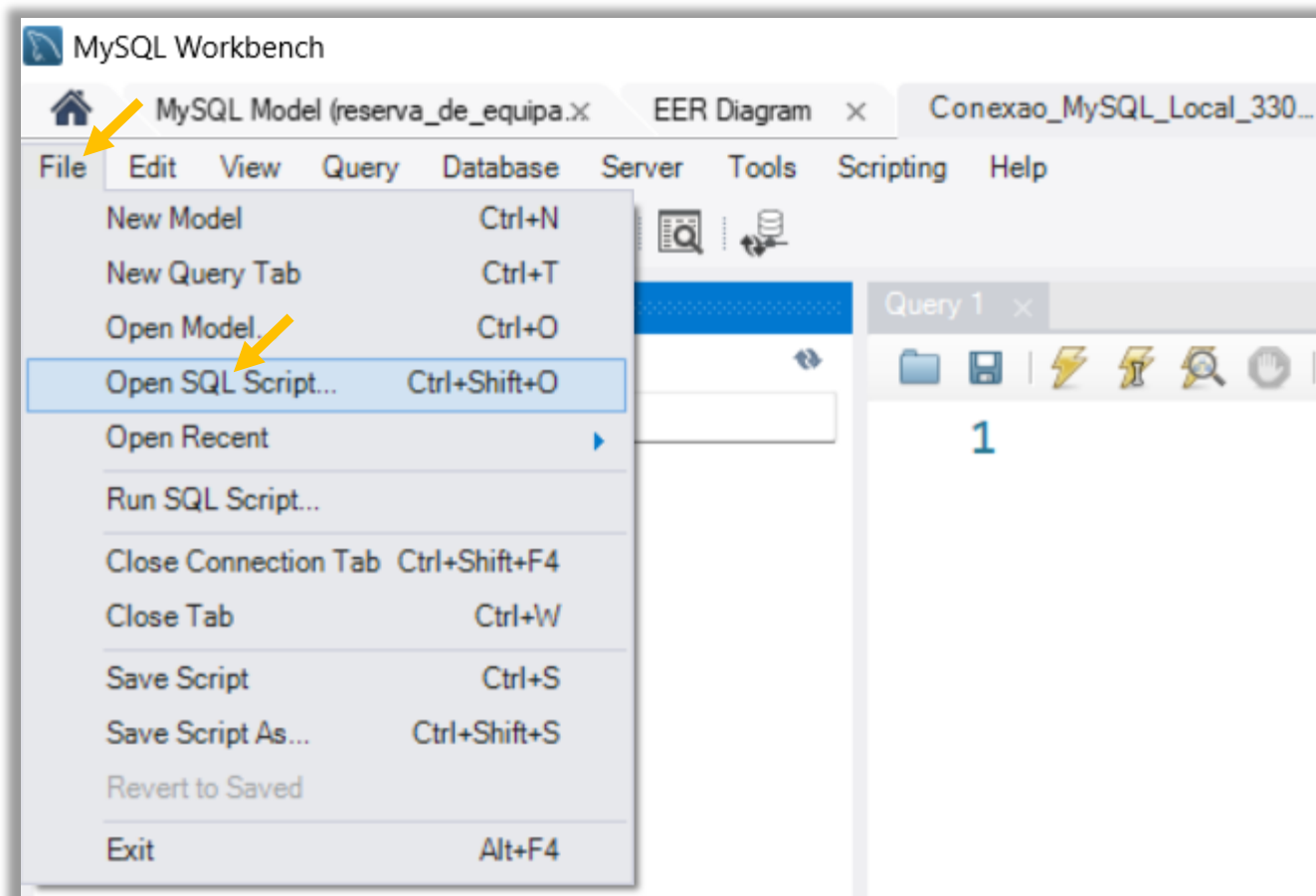
Executar o Script SQL de criação de DB



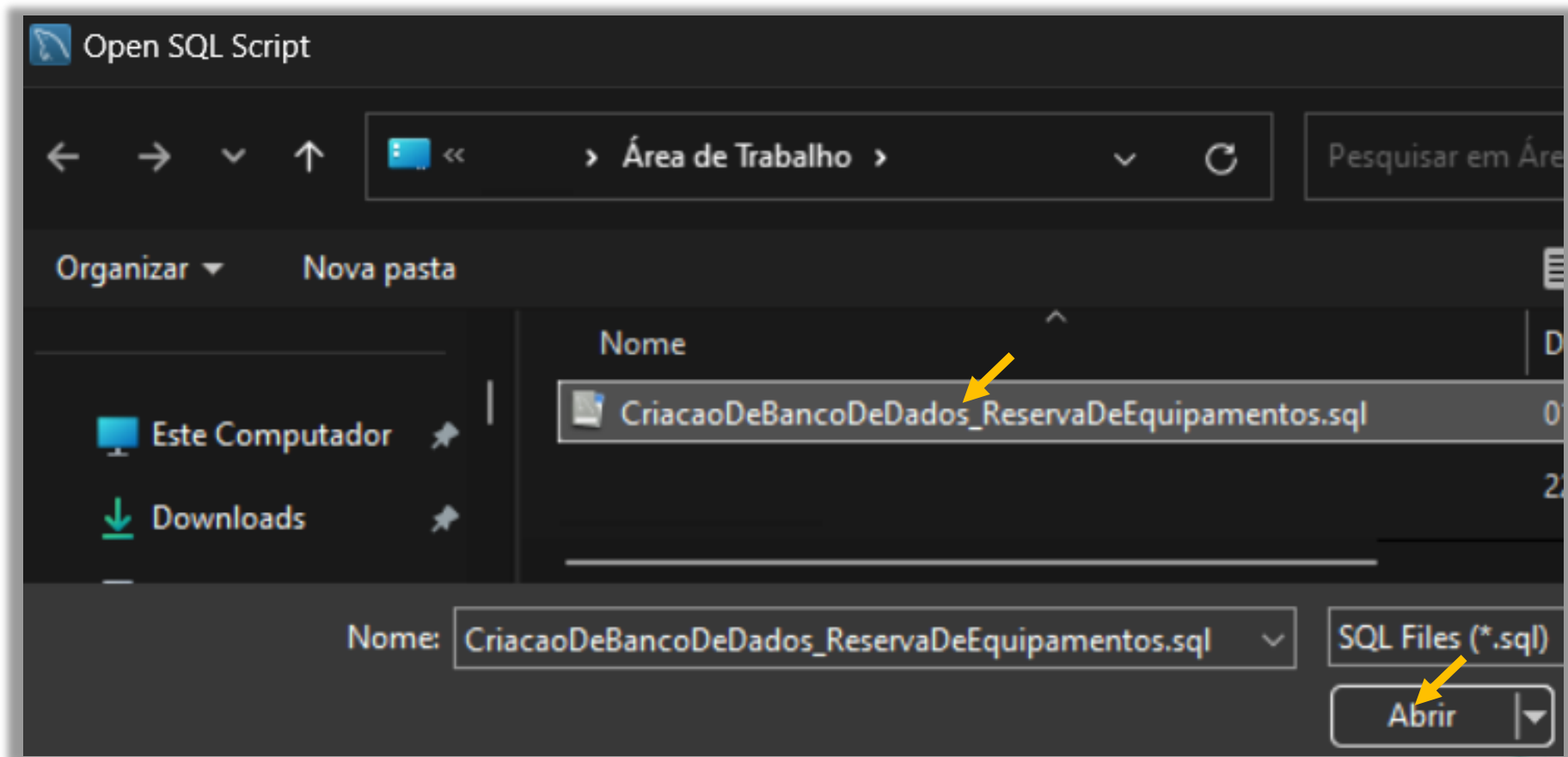
Executar o Script SQL de criação de DB

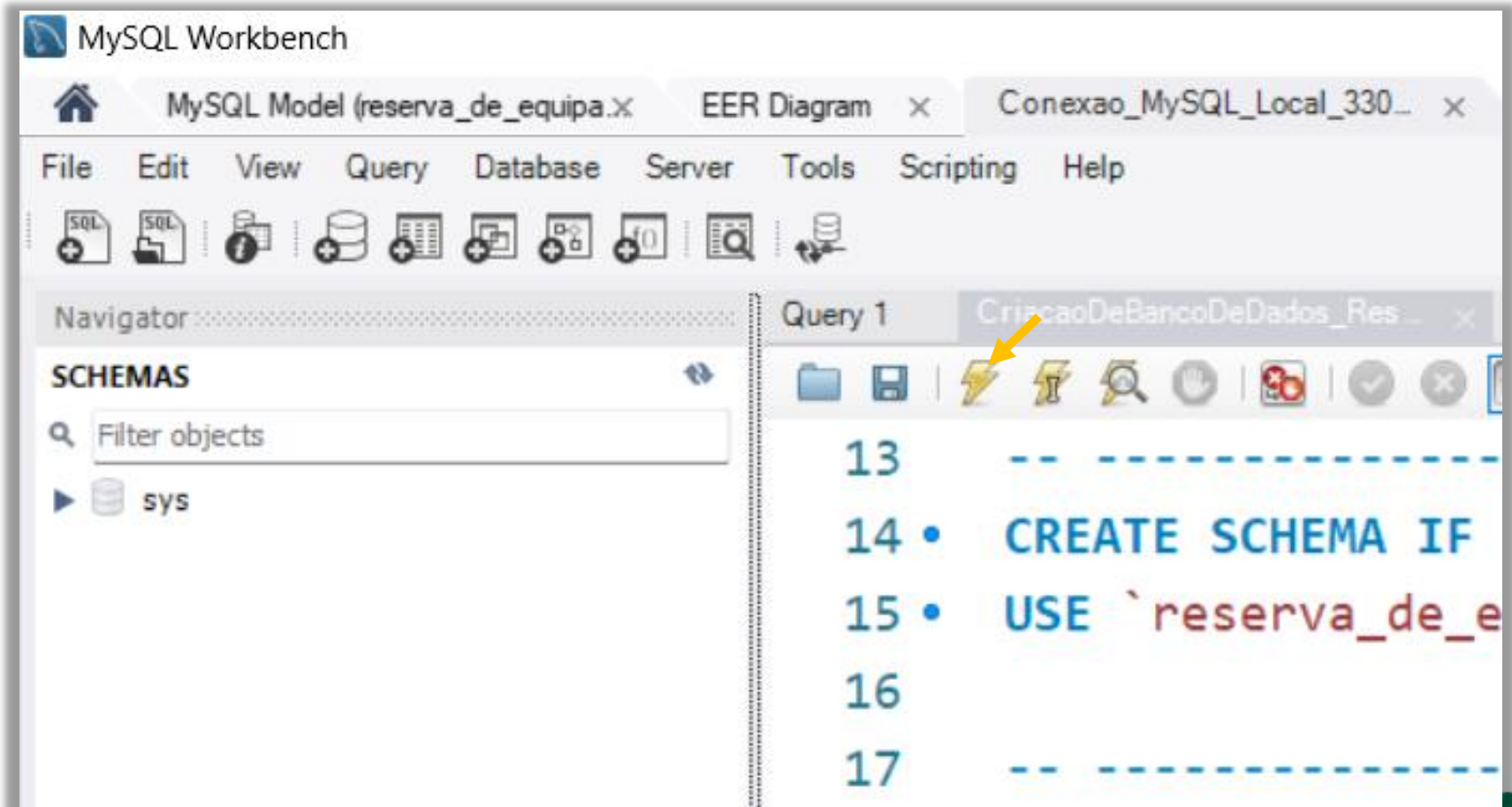


Executar o Script SQL de criação de DB



Executar o Script SQL de criação de DB



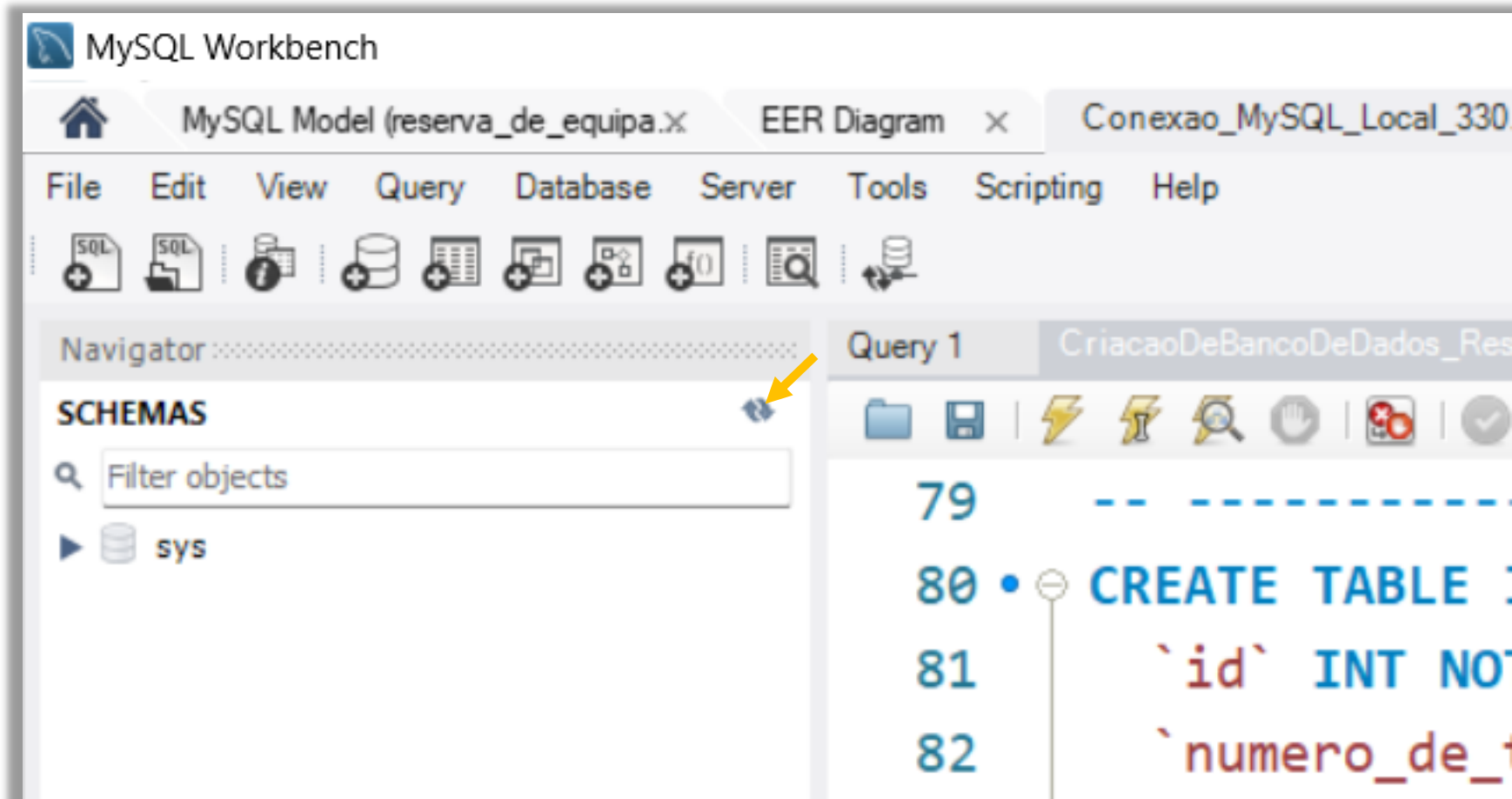


Executar o Script SQL de criação de DB

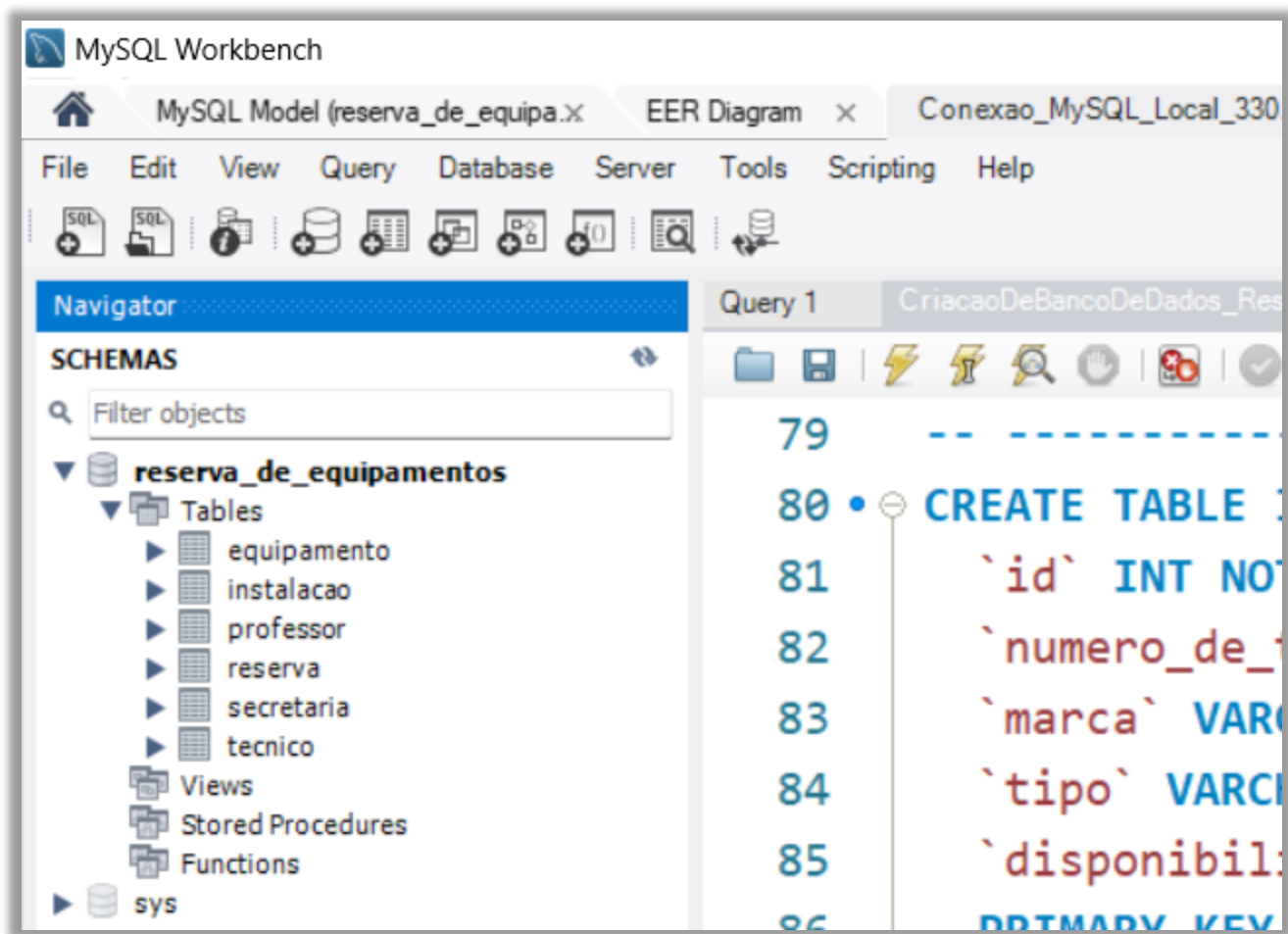
Output				
Action Output				
	#	Time	Action	Message
✓	1	17:38:41	DROP DATABASE `reserva_de_equipamentos`	6 row(s) affected
✓	2	17:43:56	SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0	0 row(s) affected
✓	3	17:43:56	SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0	0 row(s) 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 utf8mb4 COLLATE utf8mb4_0900_ai_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, `preco` DECIMAL(10,2) 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, `data` DATE NOT NULL, PRIMARY KEY (`id`), FOREIGN KEY (`professor_id`) REFERENCES `Professor` (`id`), FOREIGN KEY (`secretaria_id`) REFERENCES `Secretaria` (`id`), FOREIGN KEY (`tecnico_id`) REFERENCES `Tecnico` (`id`), FOREIGN KEY (`equipamento_id`) REFERENCES `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`), FOREIGN KEY (`reserva_id`) REFERENCES `Reserva` (`id`), FOREIGN KEY (`tecnico_id`) REFERENCES `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

```
CREATE SCHEMA IF NOT EXISTS `reserva_de Equipamentos` ;  
  
USE `reserva_de Equipamentos` ;
```



Explicação do Script SQL de criação de DB

```
CREATE TABLE IF NOT EXISTS `Professor` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `cpf` BIGINT NOT NULL,  
  `nome` VARCHAR(60) NOT NULL,  
  `email` VARCHAR(50) NULL,  
  `titulacao` VARCHAR(25) NOT NULL,  
  `rua` VARCHAR(70) NOT NULL,  
  `numero` INT NOT NULL,  
  `bairro` VARCHAR(25) NOT NULL,  
  `cidade` VARCHAR(45) NOT NULL,  
  `cep` BIGINT NOT NULL,  
  `estado` VARCHAR(40) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) VISIBLE);
```



Explicação do Script SQL de criação de DB

```
CREATE TABLE IF NOT EXISTS `Secretaria` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `cpf` BIGINT NOT NULL,  
  `nome` VARCHAR(60) NOT NULL,  
  `email` VARCHAR(50) NOT NULL,  
  `turno` VARCHAR(15) NOT NULL,  
  `rua` VARCHAR(70) NOT NULL,  
  `numero` INT NOT NULL,  
  `bairro` VARCHAR(25) NOT NULL,  
  `cidade` VARCHAR(45) NOT NULL,  
  `cep` BIGINT NOT NULL,  
  `estado` VARCHAR(40) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) VISIBLE);
```



Explicação do Script SQL de criação de DB

```
CREATE TABLE IF NOT EXISTS `Tecnico` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `cpf` BIGINT NOT NULL,  
  `nome` VARCHAR(60) NOT NULL,  
  `email` VARCHAR(50) NOT NULL,  
  `funcao` VARCHAR(25) NOT NULL,  
  `rua` VARCHAR(70) NOT NULL,  
  `numero` INT NOT NULL,  
  `bairro` VARCHAR(25) NOT NULL,  
  `cidade` VARCHAR(45) NOT NULL,  
  `cep` BIGINT NOT NULL,  
  `estado` VARCHAR(40) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) VISIBLE);
```



Explicação do Script SQL de criação de DB

```
CREATE TABLE IF NOT EXISTS `Equipamento` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `numero_de_tombamento` INT NOT NULL,  
  `marca` VARCHAR(25) NOT NULL,  
  `tipo` VARCHAR(20) NOT NULL,  
  `disponibilidade` TINYINT(1) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `numero_de_tombamento_UNIQUE`  
  (`numero_de_tombamento` ASC) VISIBLE);
```



```
CREATE TABLE IF NOT EXISTS `Reserva` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `professor_id` INT NOT NULL,  
  `secretaria_id` INT NOT NULL,  
  `local_de_instalacao` VARCHAR(30) NOT NULL,  
  `data_da_aula` DATE NOT NULL,  
  `hora_da_aula` TIME NOT NULL,  
  `data_de_solicitacao` DATE NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `FK_Reserva_Professor`  
    FOREIGN KEY (`professor_id`)  
    REFERENCES `Professor` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `FK_Reserva_Secretaria`  
    FOREIGN KEY (`secretaria_id`)  
    REFERENCES `Secretaria` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

```
CREATE TABLE IF NOT EXISTS `Instalacao` (  
  `reserva_id` INT NOT NULL,  
  `tecnico_id` INT NOT NULL,  
  `equipamento_id` INT NOT NULL,  
  `data` DATE NOT NULL,  
  PRIMARY KEY (`reserva_id`),  
  UNIQUE INDEX `reserva_id_UNIQUE` (`reserva_id` ASC) VISIBLE,  
  CONSTRAINT `FK_Instalacao_Tecnico`  
    FOREIGN KEY (`tecnico_id`)  
    REFERENCES `Tecnico` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `FK_Instalacao_Equipamento`  
    FOREIGN KEY (`equipamento_id`)  
    REFERENCES `Equipamento` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `FK_Instalacao_Reserva`  
    FOREIGN KEY (`reserva_id`)  
    REFERENCES `Reserva` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```


Exercícios

Exercícios:

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



Obrigado!

Questões?

