

Banco de Dados

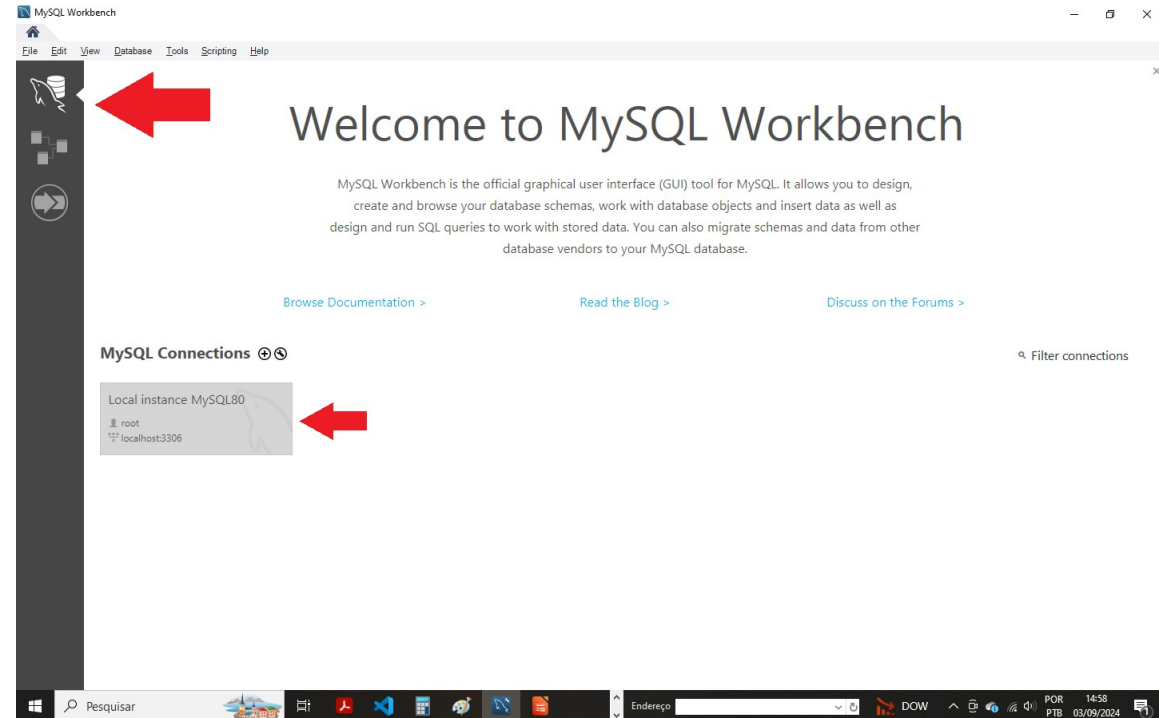
Professor Maurício Buess
mbuess@up.edu.br
github.com/mauriciobuess

Objetivo:

- Geração de scripts de Banco de Dados (MySQL)
- Introdução à linguagem DDL (Data Definition Language)
- Introdução à Linguagem DML (Data Manipulation Language)

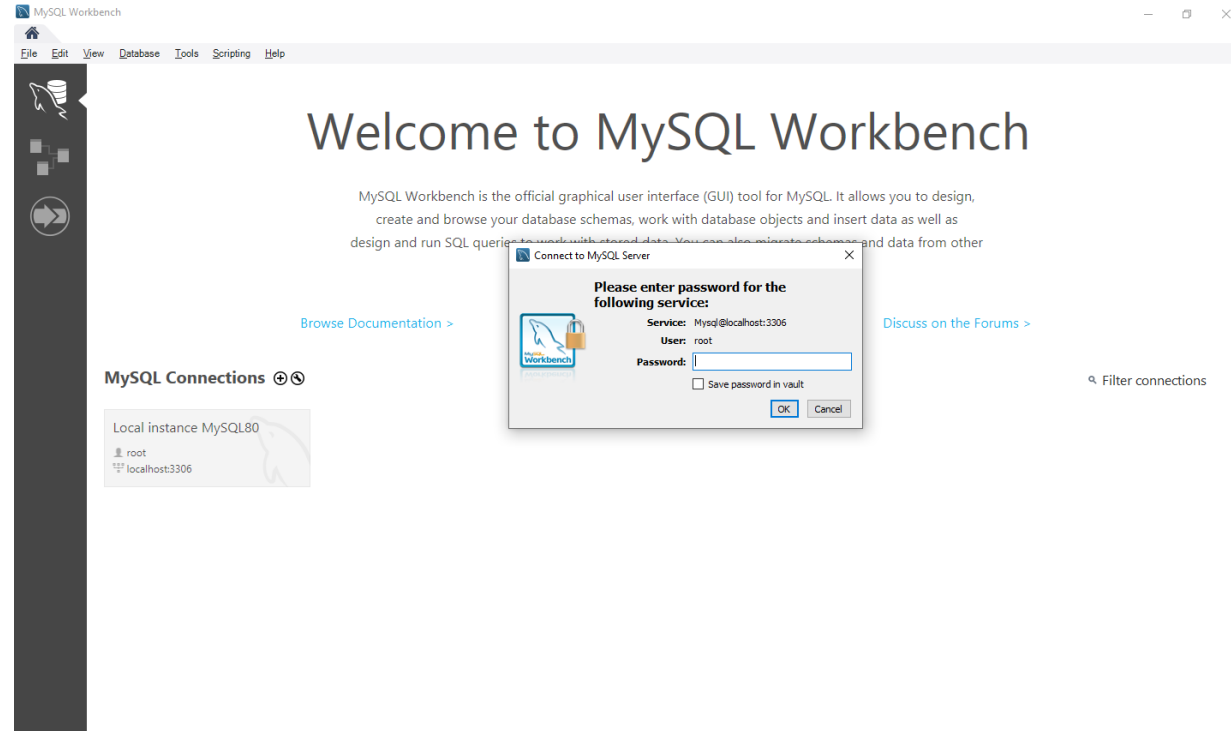
MySQL

- Inicie o MySQL;
- Escolha a opção do “golfinho”;
- Duplo clique na Instância local.



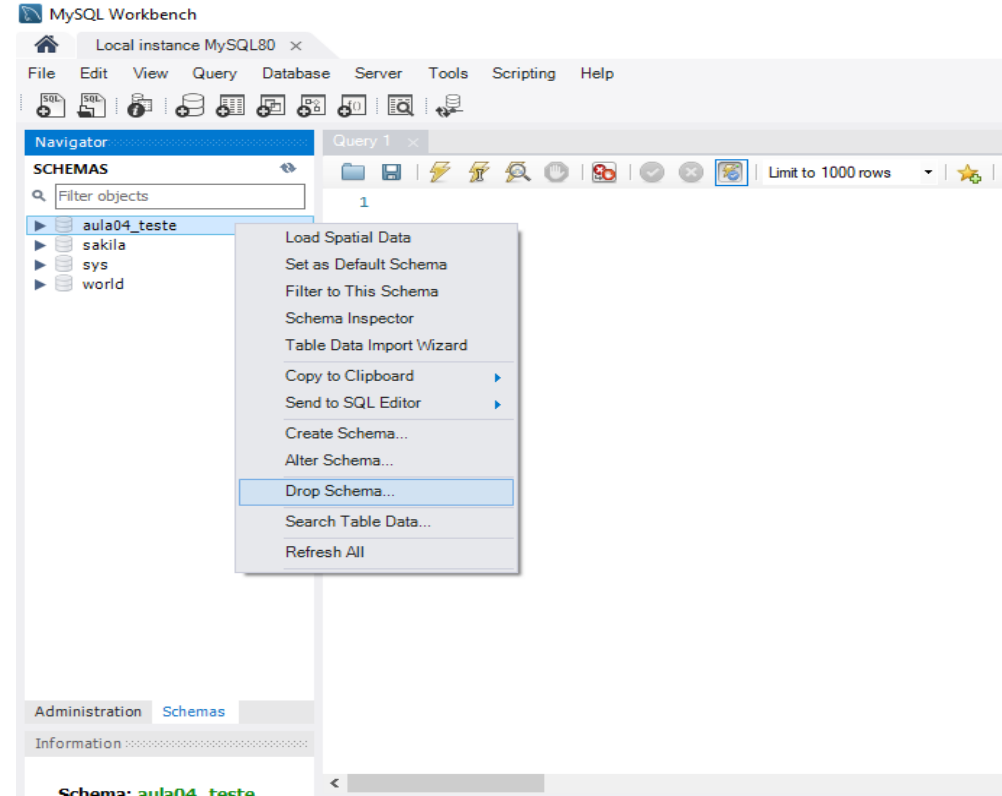
Geração de Scripts do DER MySql

- Informe a senha de acesso
- positivo



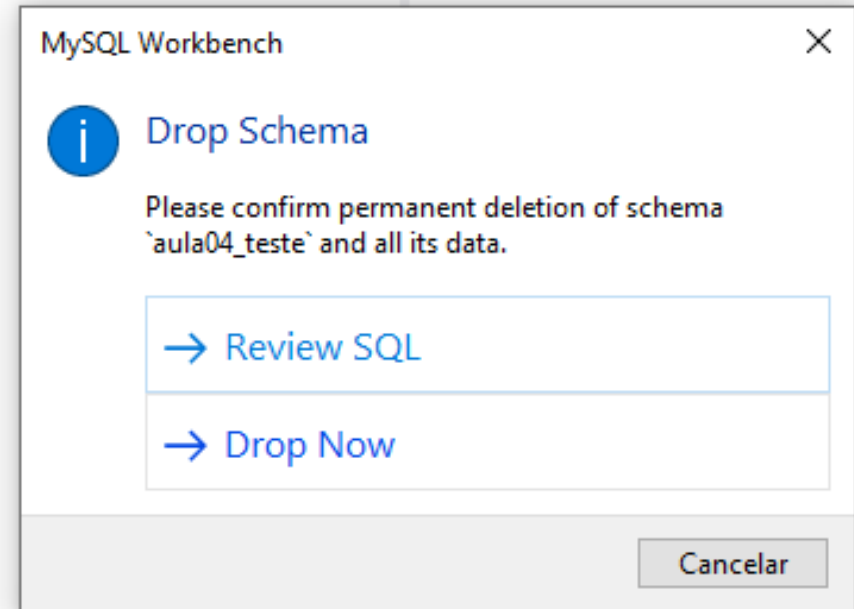
Verificação / exclusão de banco de dados

- Verifique a existência do banco de dados “aula04_teste”;
- Caso positivo, exclua-o!
- Para excluir, dê um clique com o botão direito do mouse sobre o nome do BD (aula04_teste) e escolha a opção “Drop Schema...”



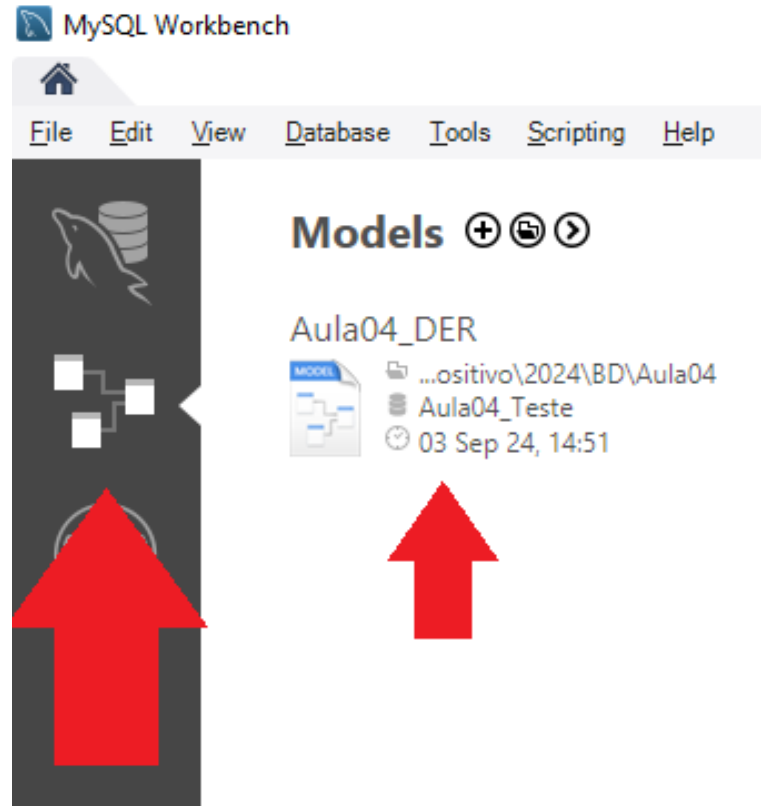
Verificação / exclusão de banco de dados

- Confirme a exclusão;
- Use “Drop Now”.
- Há outras formas de realizar essa verificação e/ou exclusão – essas outras formas serão tratadas no futuro.

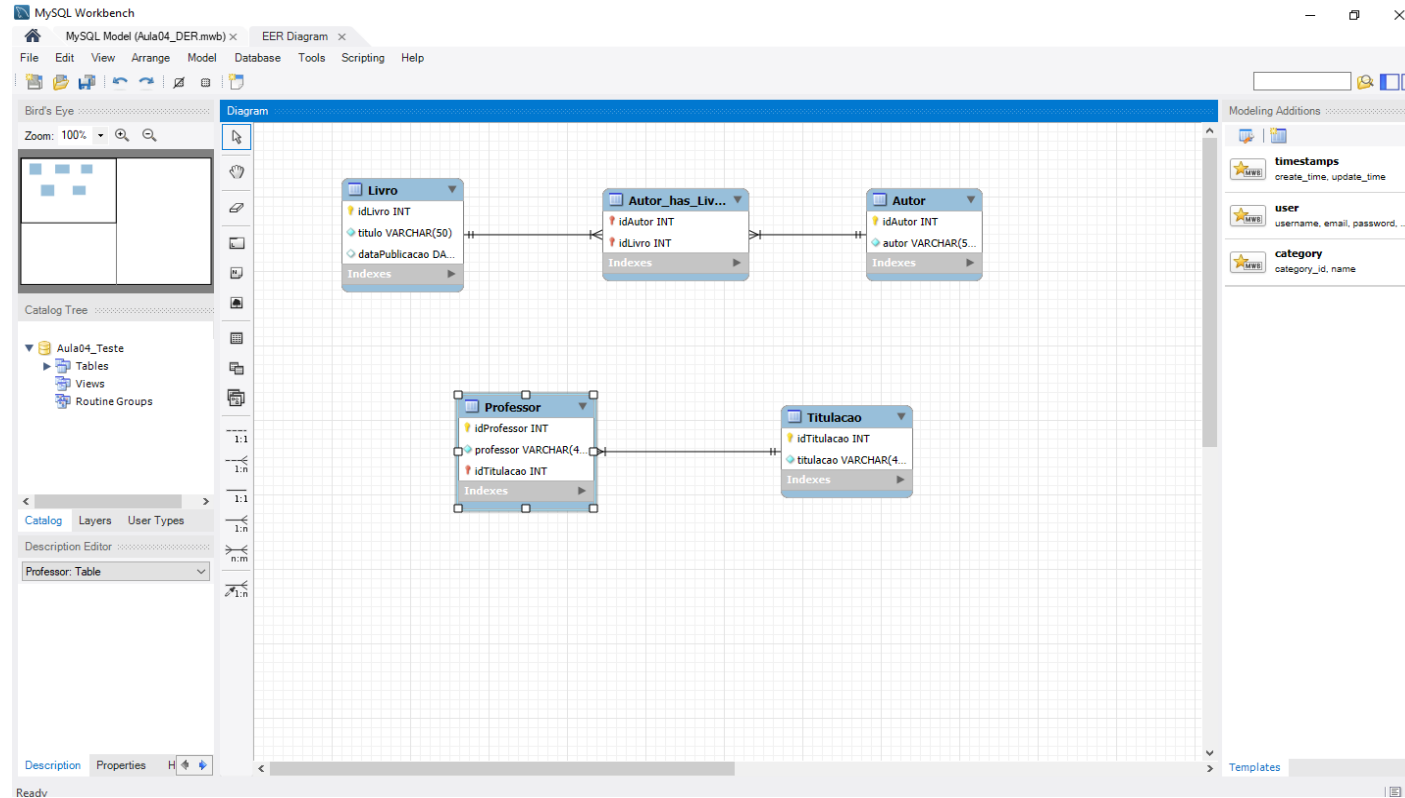


Geração do banco de dados

- É possível realizar a geração do B.D. através do DER;
- Entendemos por “gerar B.D.”
- A criação de uma área de organização de dados, e objetos (tabelas, chaves primárias, estrangeiras, views, triggers e etc).
- Feche a instância local e abra o diagrama DER

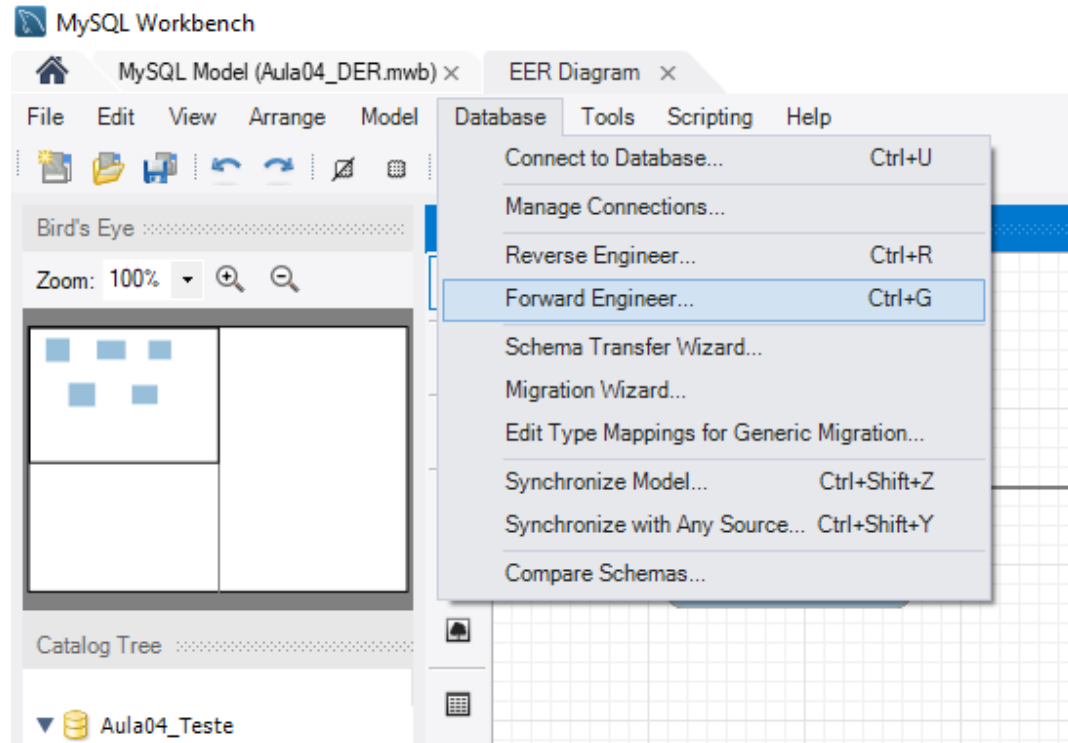


Geração do banco de dados



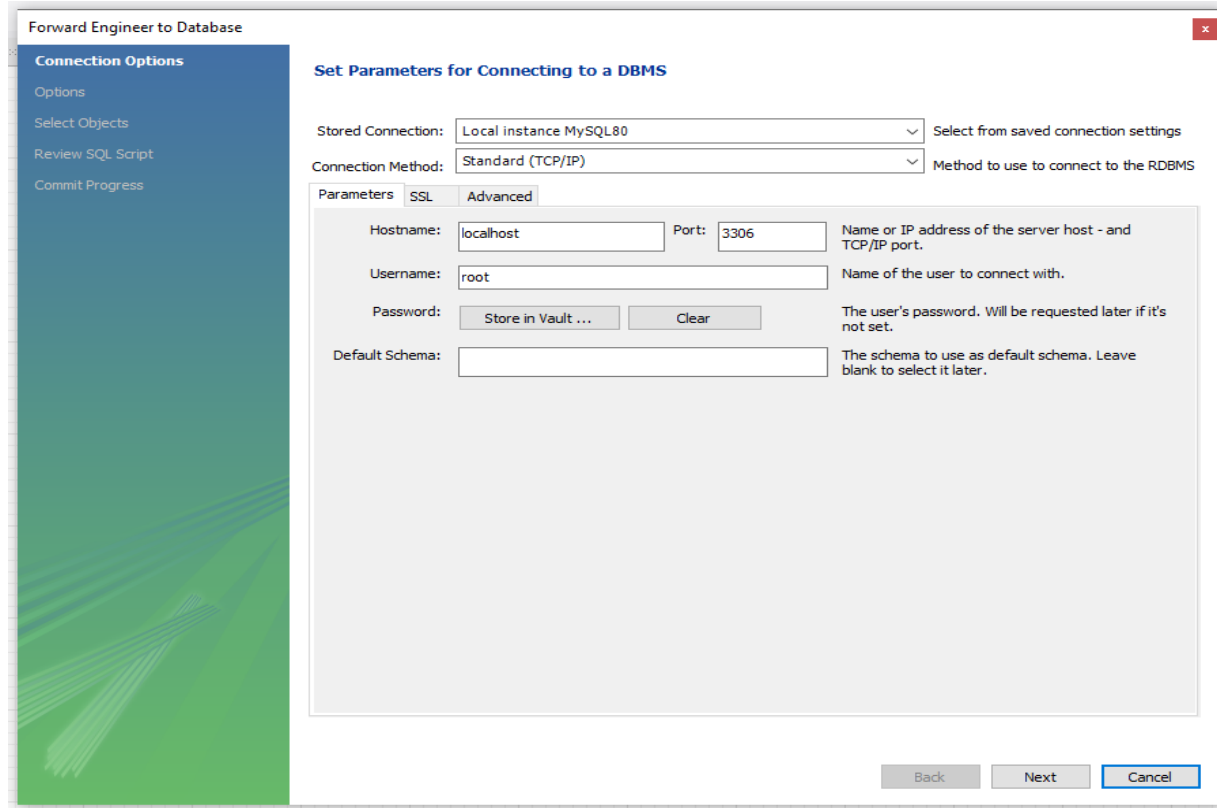
Geração do banco de dados

- Escolher a opção “Database” do menu principal;
- Escolher a opção “Forward Engineer”
- Essa opção permitirá a geração / criação de um banco de dados a partir de um DER



Geração do banco de dados

- Use a opção “Next”



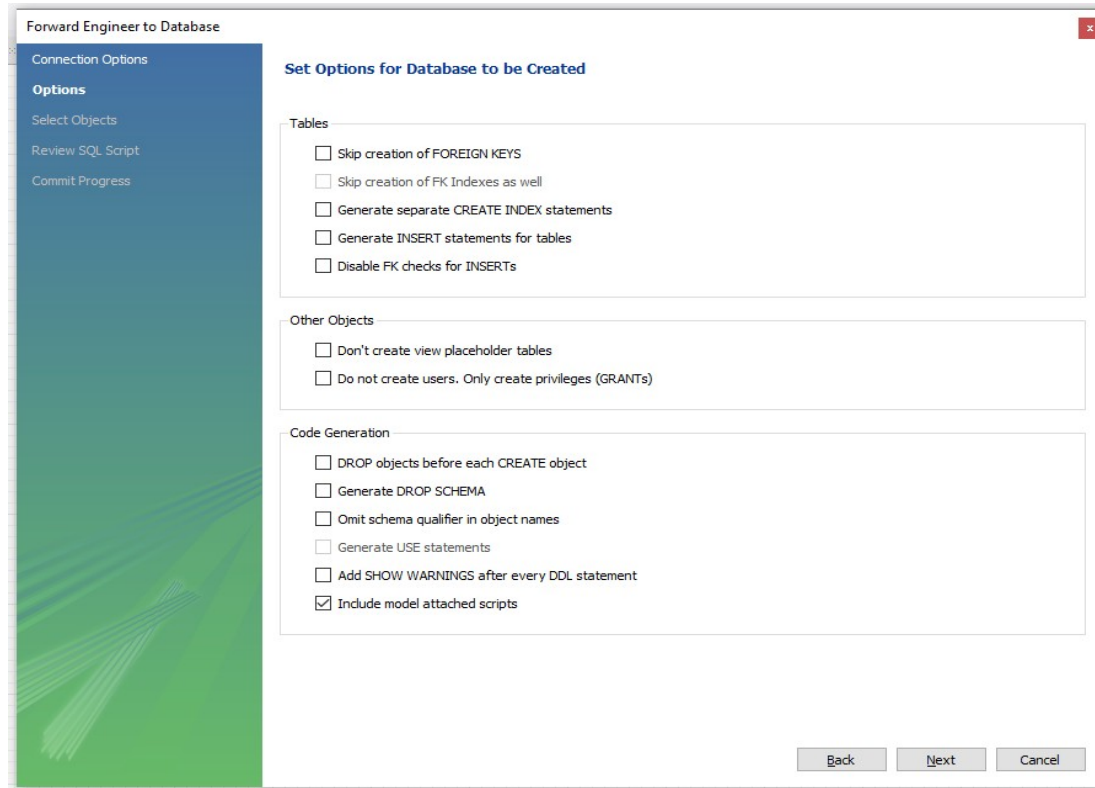
The screenshot shows a software window titled "Forward Engineer to Database". On the left is a sidebar with a blue header "Connection Options" and four menu items: "Options", "Select Objects", "Review SQL Script", and "Commit Progress". The main area is titled "Set Parameters for Connecting to a DBMS" and contains the following fields and controls:

- Stored Connection:** A dropdown menu showing "Local instance MySQL80". To its right is the text "Select from saved connection settings".
- Connection Method:** A dropdown menu showing "Standard (TCP/IP)". To its right is the text "Method to use to connect to the RDBMS".
- Below these are three tabs: "Parameters" (selected), "SSL", and "Advanced".
- Under the "Parameters" tab:
 - Hostname:** A text box containing "localhost". To its right is a **Port:** text box containing "3306". To the right of the port box is the text "Name or IP address of the server host - and TCP/IP port."
 - Username:** A text box containing "root". To its right is the text "Name of the user to connect with."
 - Password:** A text box followed by two buttons: "Store in Vault ..." and "Clear". To the right is the text "The user's password. Will be requested later if it's not set."
 - Default Schema:** A text box. To its right is the text "The schema to use as default schema. Leave blank to select it later."

At the bottom right of the window are three buttons: "Back", "Next", and "Cancel". The "Next" button is highlighted with a blue border.

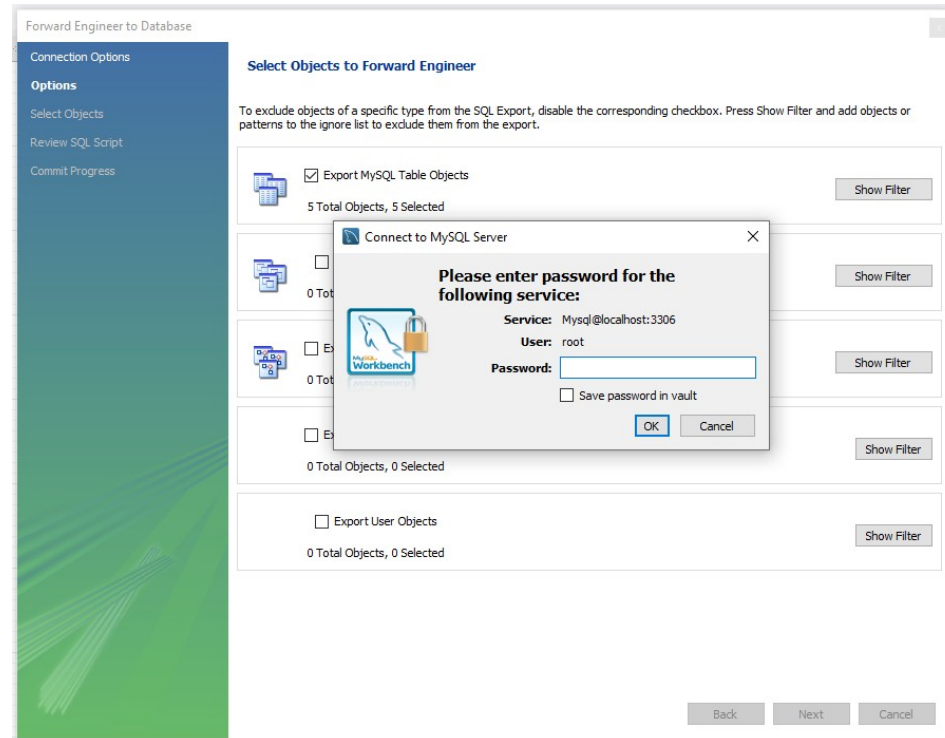
Geração do banco de dados

- Use a opção “Next”



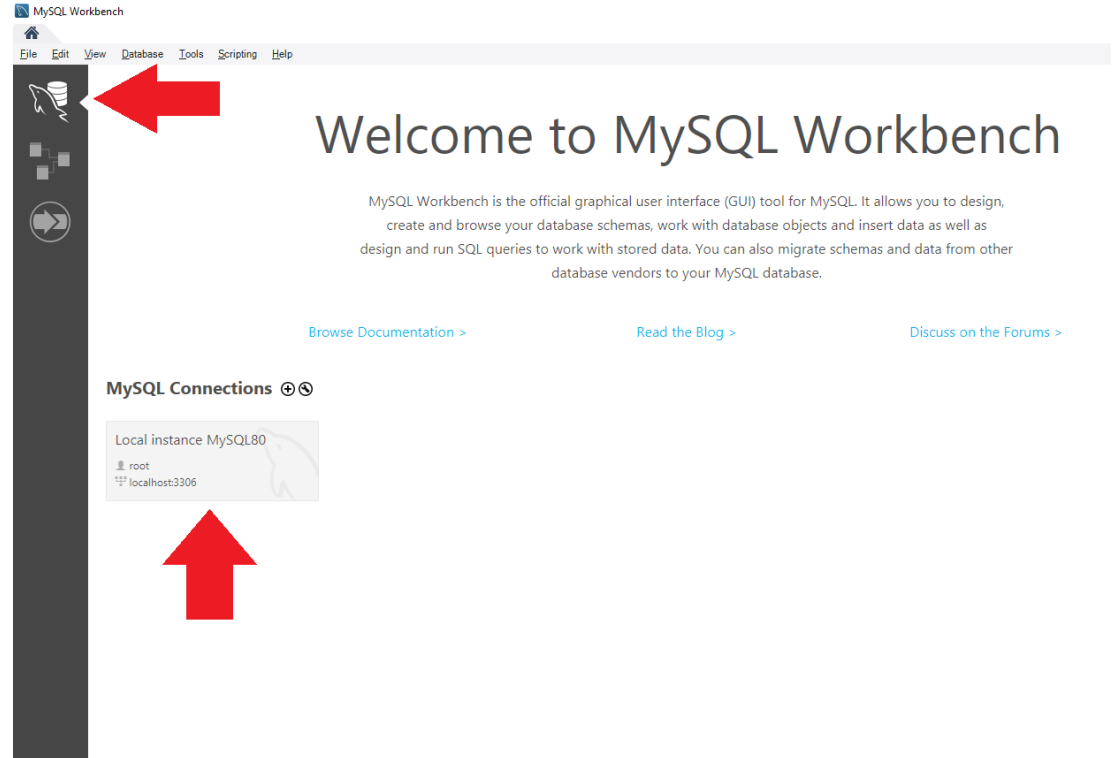
Geração do banco de dados

- Informe a senha de acesso;
 - Pressione “Ok”;
 - Escolha o botão “Next”;
 - Novamente o “Next”.
- Processo concluído
- Feche a tela do DER e volte para “Home”

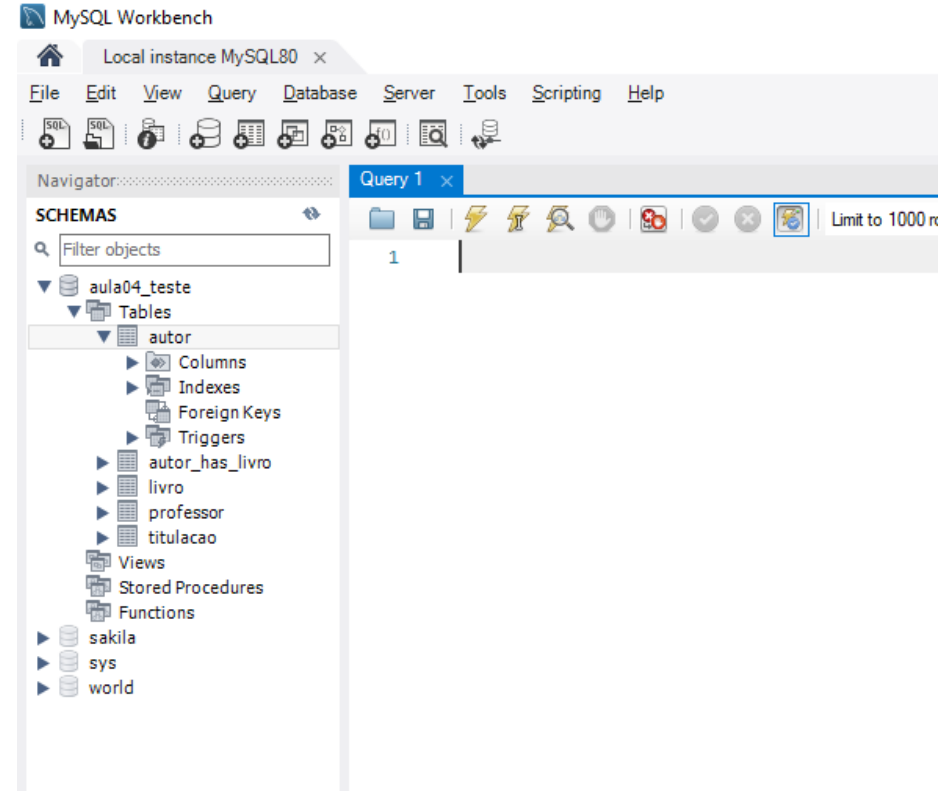
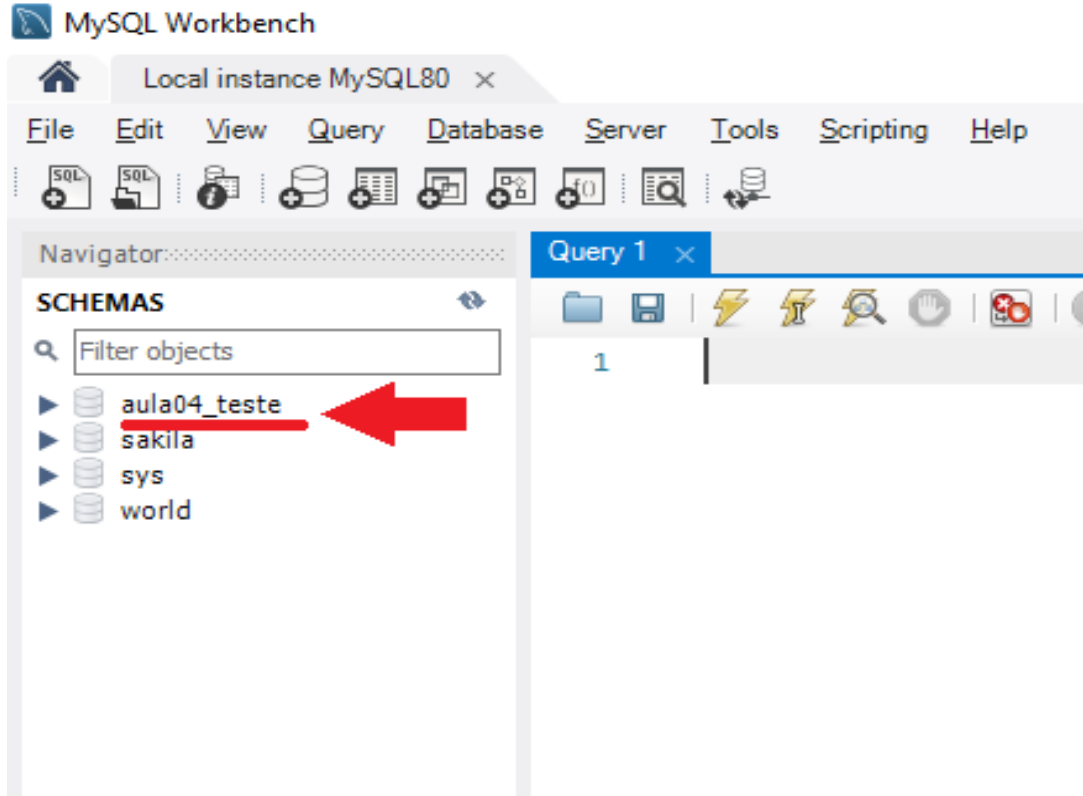


Verificação da Geração do banco de dados

- Retorne à primeira tela do MySql
- Escolhe a opção de banco de dados
- Duplo clique na conexão
- Informe a senha, caso seja pedido



Verificação da Geração do banco de dados



Linguagens do MySql

- **SQL** é a linguagem principal utilizada para interagir com o MySQL. Ela inclui várias sub-linguagens e comandos que permitem a definição, manipulação e controle dos dados no banco de dados.
- **DDL (Data Definition Language)**: Para definir e modificar a estrutura dos objetos do banco de dados.
 - **CREATE**: Cria novas tabelas, índices, views, etc.
 - **ALTER**: Modifica a estrutura de tabelas e outros objetos.
 - **DROP**: Remove tabelas, índices, views, etc.

Linguagens do MySql

- **DML (Data Manipulation Language):** Para manipular os dados armazenados no banco de dados.
 - SELECT: Consulta dados.
 - INSERT: Adiciona novos dados.
 - UPDATE: Modifica dados existentes.
 - DELETE: Remove dados.
- **DCL (Data Control Language):** Para controlar permissões e acesso aos dados.
 - GRANT: Concede permissões a usuários.
 - REVOKE: Revoga permissões de usuários.

Linguagens do MySql

- **TCL (Transaction Control Language):** Para gerenciar transações no banco de dados.
 - COMMIT: Salva todas as alterações feitas na transação.
 - ROLLBACK: Desfaz as alterações feitas na transação.
 - SAVEPOINT: Define um ponto de salvamento dentro da transação.

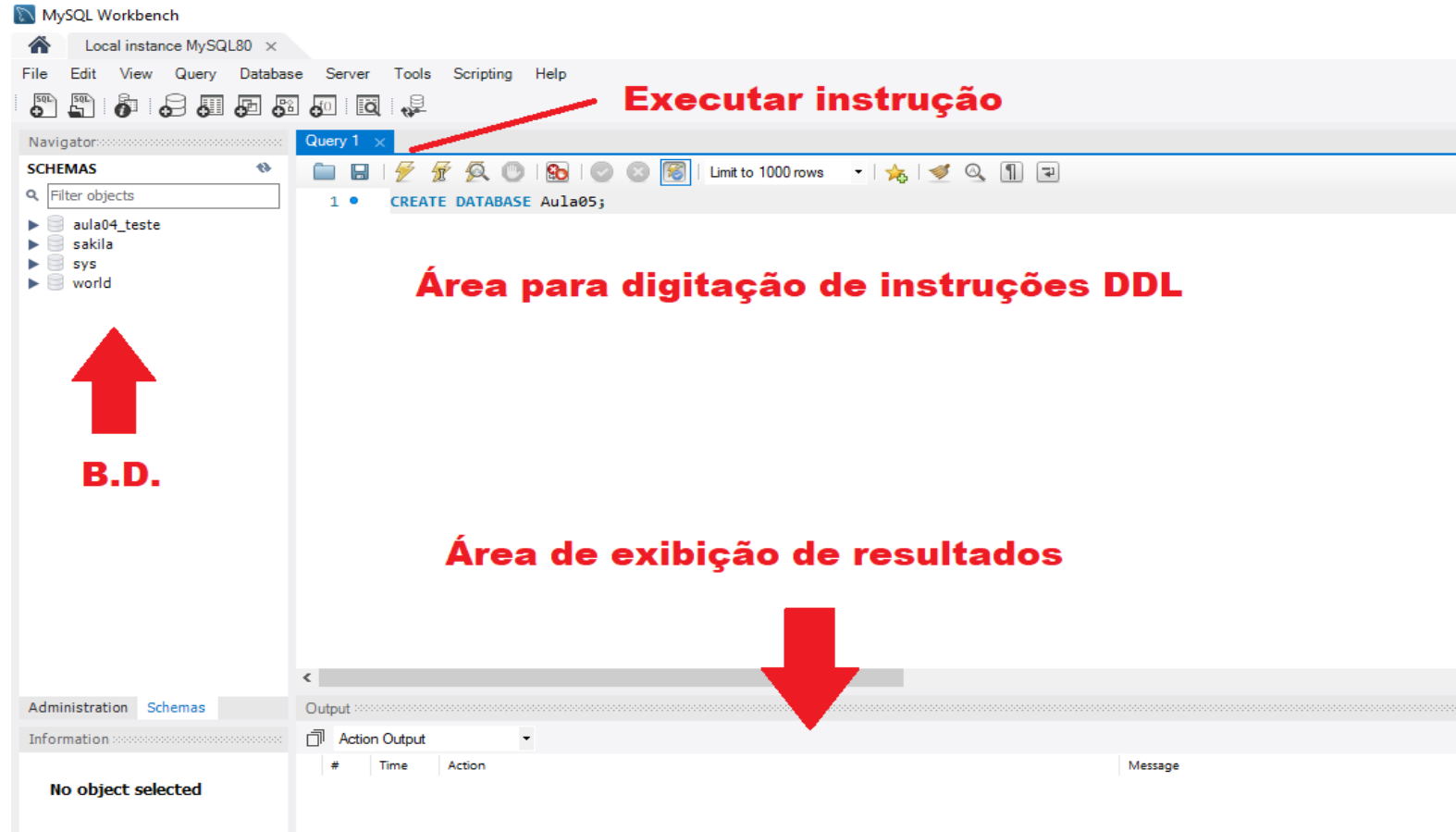
DDL (Data Definition Language)

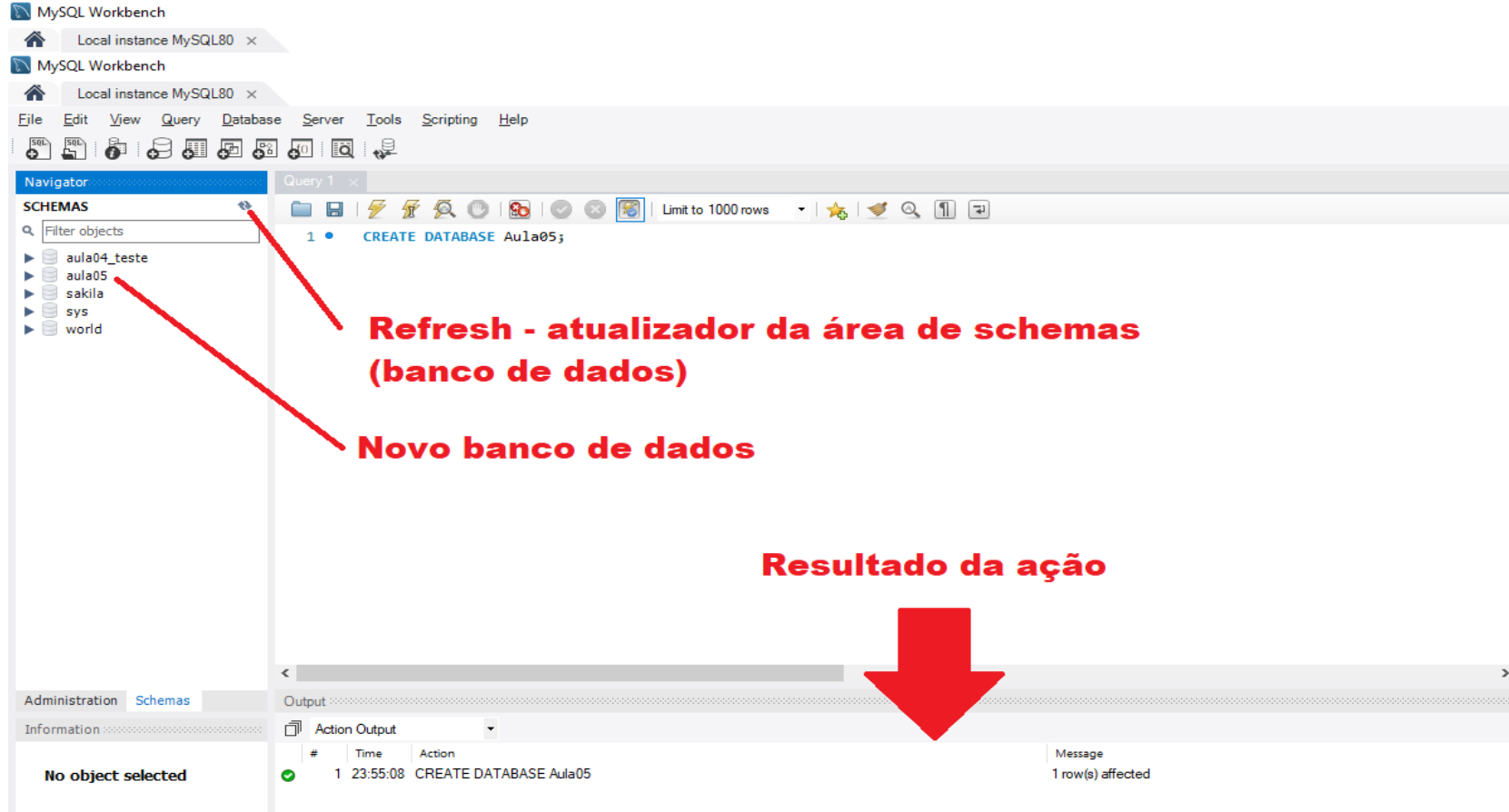
- Criando um Banco de Dados
 - Antes de criar tabelas e outros objetos, é necessário criar um banco de dados.
 - Para isso se usa o comando CREATE DATABASE:

```
CREATE DATABASE my_database;
```

- my_database → Nome do banco de dados.
- Exemplo:

```
CREATE DATABASE Aula05;
```





MySQL Workbench

Local instance MySQL80 x

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- aula04_teste
- aula05
- sakila
- sys
- world

Query 1 x

1 • CREATE DATABASE Aula05;

Limit to 1000 rows

Refresh - atualizador da área de schemas (banco de dados)

Novo banco de dados

Resultado da ação

Administration Schemas

Information

No object selected

Output

Action Output

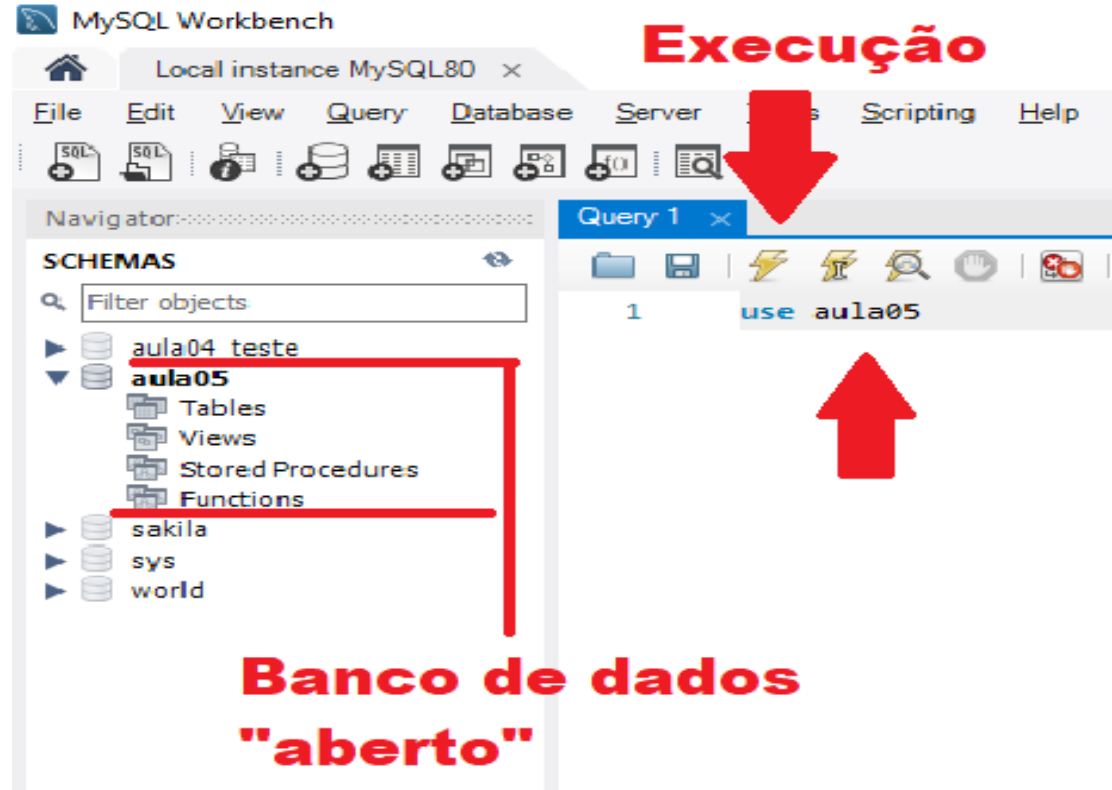
#	Time	Action	Message
✓ 1	23:55:08	CREATE DATABASE Aula05	1 row(s) affected

DDL (Data Definition Language)

- Após criar o banco de dados, é preciso selecioná-lo para criar tabelas e outros objetos;
- Esse processo de selecionar o banco de dados é conhecido como “abrir o banco de dados”;

USE aula05

- Após a digitação do comando pressionar a tecla de execução.



DDL (Data Definition Language)

- Outra ação importante é a criação de tabela;
- A criação de tabela deve ocorrer com o banco de dados aberto:

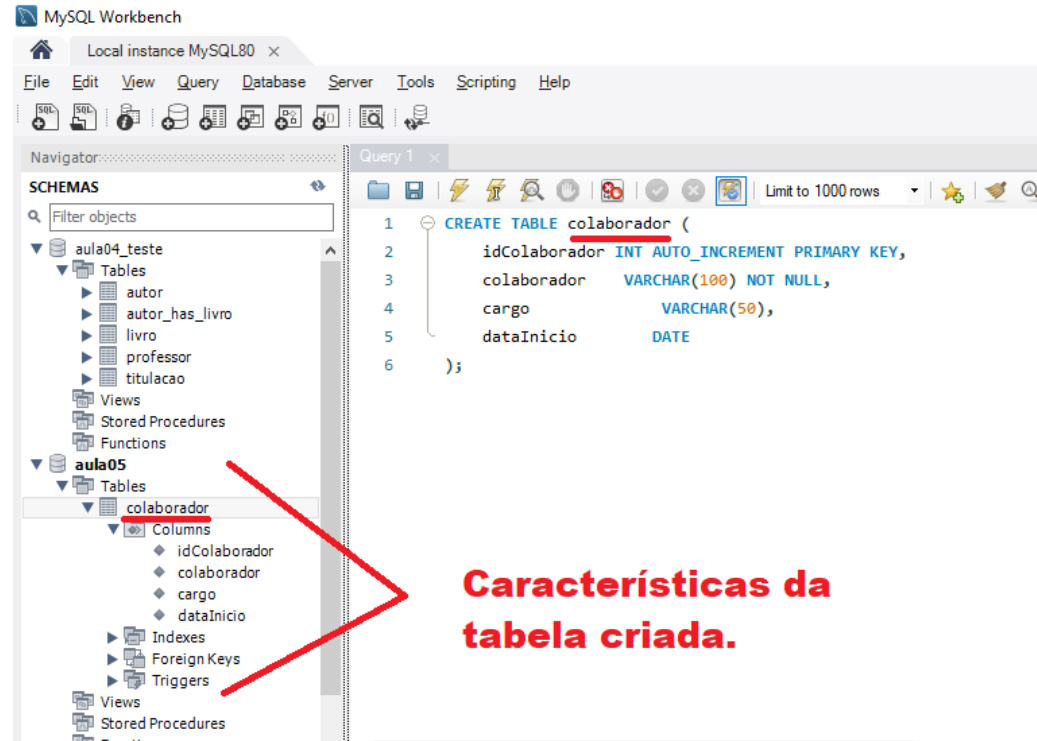
CREATE TABLE

- Exemplo:

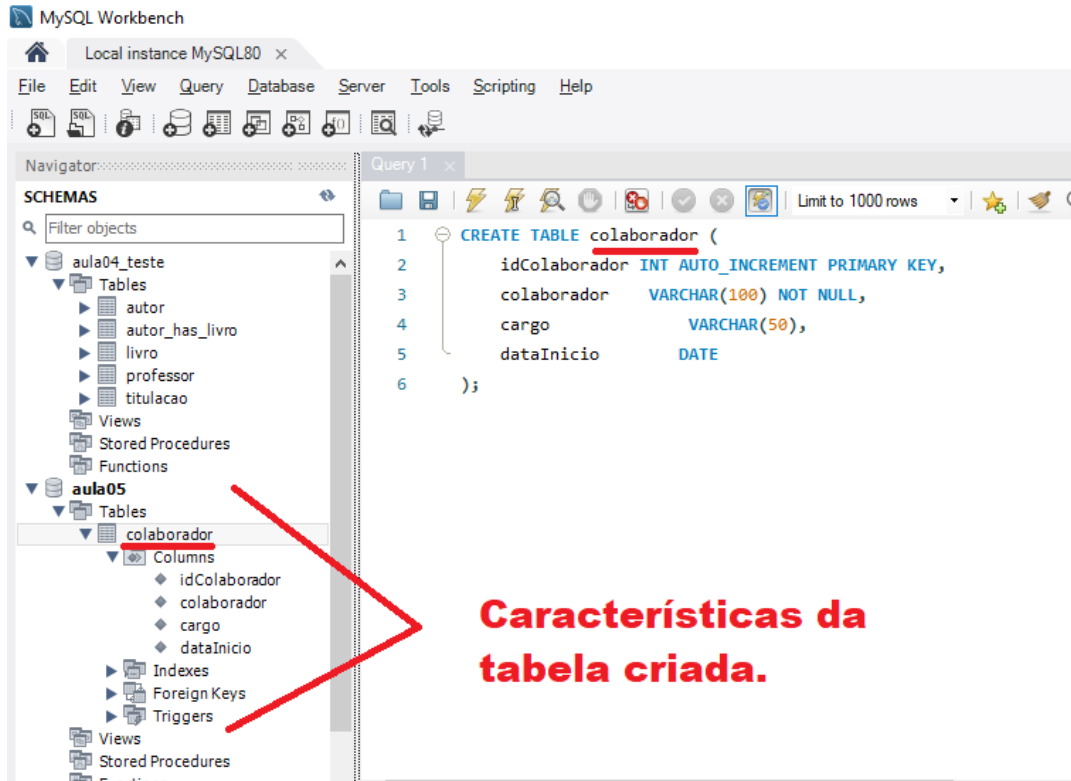
```
CREATE TABLE colaborador (  
    idColaborador INT AUTO_INCREMENT PRIMARY KEY,  
    colaborador   VARCHAR(100) NOT NULL,  
    cargo         VARCHAR(50),  
    dataInicio    DATE  
)
```

DDL (Data Definition Language)

- Após execução do comando 'Create Table':
- Expanda o grupo Tables
- Localize a tabela 'colaborador'
- Expanda o grupo Columns
- Compare o resultado com o comando



DDL (Data Definition Language)



- idColaborador: Coluna de identificação do empregado, tipo INT com auto incremento e chave primária.
- colaborador: Nome do empregado, tipo VARCHAR(100) e não pode ser nulo.
- cargo: Cargo do empregado, tipo VARCHAR(50).
- dataInicio: Data de contratação, tipo DATE.

DDL (Data Definition Language)

- Apagar uma tabela: 'Drop table'
- Comando para apagar a tabela 'colaborador';

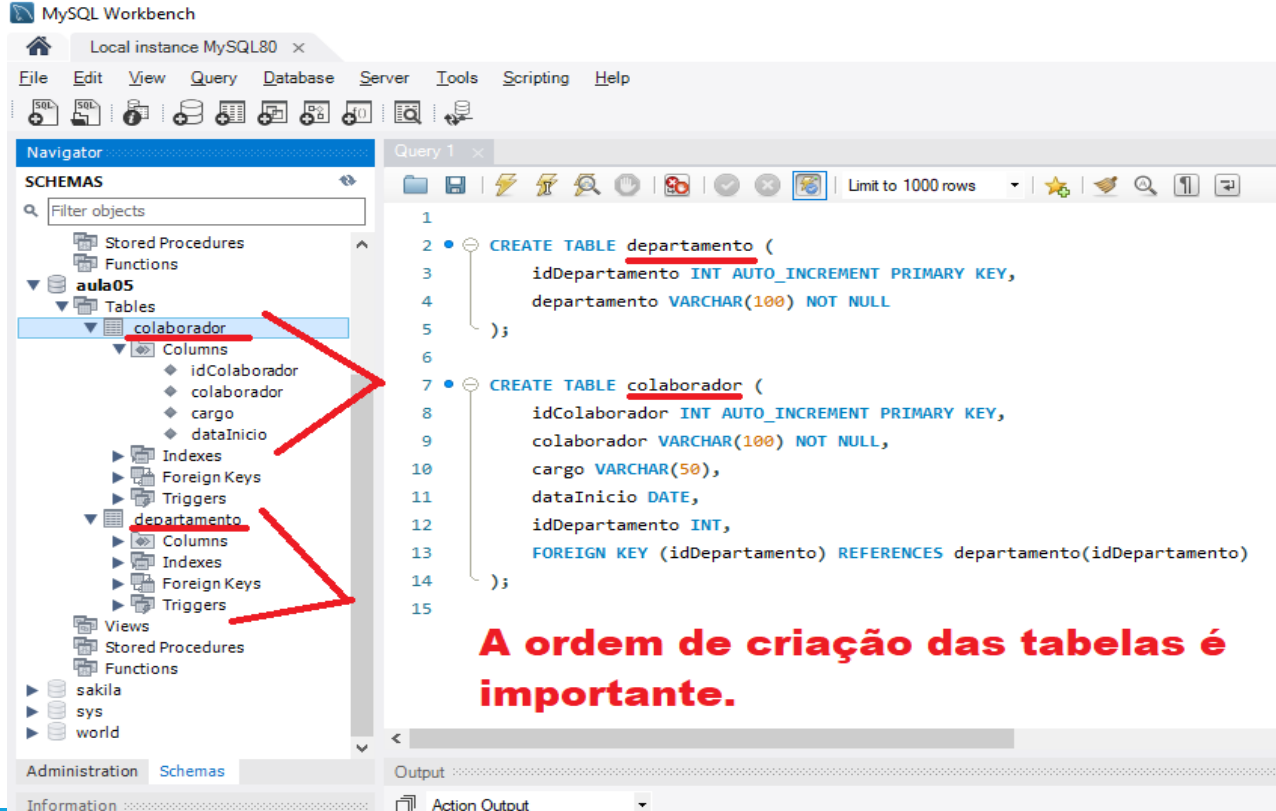
Drop table colaborador

- Pressione o botão de execução e verifique o banco de dados;
- Criação de tabela com chave estrangeira:

```
CREATE TABLE departamento (  
    idDepartamento INT AUTO_INCREMENT PRIMARY KEY,  
    departamento VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE colaborador (  
    idColaborador INT AUTO_INCREMENT PRIMARY KEY,  
    colaborador VARCHAR(100) NOT NULL,  
    cargo VARCHAR(50),  
    dataInicio DATE,  
    idDepartamento INT,  
    FOREIGN KEY (idDepartamento) REFERENCES  
    departamento(idDepartamento)  
);
```

DDL (Data Definition Language)



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Stored Procedures

Functions

aula05

Tables

colaborador

Columns

idColaborador

colaborador

cargo

dataInicio

Indexes

Foreign Keys

Triggers

departamento

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

sakila

sys

world

Query 1 x

Limit to 1000 rows

```
1
2 CREATE TABLE departamento (
3     idDepartamento INT AUTO_INCREMENT PRIMARY KEY,
4     departamento VARCHAR(100) NOT NULL
5 );
6
7 CREATE TABLE colaborador (
8     idColaborador INT AUTO_INCREMENT PRIMARY KEY,
9     colaborador VARCHAR(100) NOT NULL,
10    cargo VARCHAR(50),
11    dataInicio DATE,
12    idDepartamento INT,
13    FOREIGN KEY (idDepartamento) REFERENCES departamento(idDepartamento)
14 );
15
```

A ordem de criação das tabelas é importante.

Administration Schemas

Information Action Output

DDL (Data Definition Language)

- Para modificar a estrutura de tabelas existentes utiliza-se o ALTER TABLE

Exemplo: Adicionar a coluna e-mail na tabela colaborador

```
ALTER TABLE colaborador ADD email VARCHAR(100);
```

- Adiciona uma nova coluna chamada email do tipo VARCHAR(100) à tabela colaborador.

```
ALTER TABLE colaborador MODIFY cargo VARCHAR(100);
```

- Modifica a coluna position para ter um comprimento máximo de 100 caracteres.

DDL (Data Definition Language)

- Renomear uma Coluna

```
ALTER TABLE colaborador RENAME COLUMN dataInicio TO data_Inicio;
```

- Renomeia a coluna dataInicio para data_Inicio.

- Excluir uma Coluna

```
ALTER TABLE colaborador DROP COLUMN email;
```

- Remove a coluna email da tabela colaborador.

DML (Data Manipulation Language):

- **Linguagem de manipulação de dados**
 - INSERT é usado para adicionar novos registros a uma tabela no banco de dados.
 - Várias formas de usar o INSERT, dependendo de como você deseja adicionar dados.

```
INSERT INTO colaborador (colaborador, cargo, data_Inicio)
VALUES ('Rolando Caio da Rocha', 'Engenheiro de Software', '2024-09-01');
```

```
INSERT INTO colaborador (colaborador, cargo)
VALUES ('Fiel Augusto dos Anjos', 'Gerente de Projeto');
```

- Inserir mais cinco colaboradores na tabela colaborador.

DML (Data Manipulation Language):

- **Linguagem de manipulação de dados**
 - INSERT é usado para adicionar novos registros a uma tabela no banco de dados.
 - Várias formas de usar o INSERT, dependendo de como você deseja adicionar dados.

```
INSERT INTO colaborador (colaborador, cargo, data_Inicio)  
VALUES ('Rolando Caio da Rocha', 'Engenheiro de Software', '2024-09-01');
```

```
INSERT INTO colaborador (colaborador, cargo)  
VALUES ('Fiel Augusto dos Anjos', 'Gerente de Projeto');
```

- Inserir mais cinco colaboradores na tabela colaborador.

DML (Data Manipulation Language):

- **SELECT** → usado para consultar (recuperar) dados de uma ou mais tabelas.
- Pode especificar quais colunas se deseja, aplicar filtros, ordenar os resultados e muito mais.

Exemplo: Selecionar Todas as Colunas

```
SELECT * FROM colaborador;
```

- **SELECT ***: Seleciona todas as colunas da tabela colaborador.
- **FROM colaborador**: Indica a tabela da qual os dados serão recuperados.

DML (Data Manipulation Language):

- **SELECT** → usado para consultar (recuperar) dados de uma ou mais tabelas.
- Pode especificar quais colunas se deseja, aplicar filtros, ordenar os resultados e muito mais.

Exemplo: Selecionar Todas as Colunas

```
SELECT * FROM colaborador;
```

- **SELECT ***: Seleciona todas as colunas da tabela colaborador.
- **FROM colaborador**: Indica a tabela da qual os dados serão recuperados.

DML (Data Manipulation Language):

- Seleciona apenas as colunas colaborador e cargo da tabela colaborador:

```
SELECT name, position FROM employees;
```

- Filtrar Resultados com WHERE

```
SELECT * FROM colaborador WHERE cargo = 'Engenheiro de Software'
```

- WHERE cargo = 'Engenheiro de Software': Filtra os registros para mostrar apenas aqueles em que a cargo é igual a 'Engenheiro de Software'.

```
SELECT * FROM colaborador WHERE data_Inicio > '2024-01-01'
```

```
select * from colaborador where data_inicio IS NULL;
```

DML (Data Manipulation Language):

- Ordenar Resultados com ORDER BY

SELECT * FROM colaborador ORDER BY data_Inicio;

- ORDER BY data_Inicio: Ordena os resultados pela coluna data_Inicio em ordem crescente.

SELECT * FROM colaborador ORDER BY data_Inicio DESC;

- ORDER BY data_Inicio: Ordena os resultados pela coluna data_Inicio em ordem decrescente.

DML (Data Manipulation Language):

- Limitar Resultados com a Cláusula LIMIT

SELECT * FROM colaborador LIMIT 5;

- LIMIT 5: Retorna apenas os primeiros 5 registros da consulta.
- Agregar Dados
- SELECT COUNT(*) AS num_colaboradores FROM colaborador;
 - COUNT(*): Conta o número total de registros na tabela employees.
 - AS num_colaboradores: Dá um nome à coluna resultante.

Exercícios

- 1) Incluir os seguintes departamentos na tabela 'departamento':
- 2) Vendas, Almoxarifado, Financeiro, Recursos Humanos, Produção, Faturamento
- 3) Inserir os seguintes colaboradores:
 - Ana Braga Maria - Recursos humanos - iniciou em '2001-04-01'
 - Alan Brado da Rocha - Produção - iniciou em '2022-12-10'
 - Tomaz Cunha - Faturamento - iniciou em '2023-08-31'
 - Adalberto Pinto - Almoxarifado - iniciou em '2021-10-15'
 - Delfin Safrá - Financeiro - Iniciou em '2019-11-05'
 - Rolando Mineiro - Vendas - Iniciou em '2001-03-15'
 - Rosinha Linda - Vendas - Iniciou em '2005-03-01'
 - Gastão Neto - Financeiro - Iniciou em '2024-02-01'

Exercícios

- 3) Apresentar todos os departamentos da empresa:
- 4) Mostrar todos os funcionários da empresa ordenados alfabeticamente pelo nome
- 5) Contar quantos funcionários são do departamento de vendas;
- 6) Mostrar quem é o funcionário mais antigo;
- 7) Apresentar os três mais antigos funcionários;
- 8) Quais são os funcionários que não estão lotados em um setor da empresa?
- 9) Quem é o funcionário mais novo da empresa?
- 10) Listar os colaboradores ordenados pelo departamento e pelos nome.