

# **Bacharelado em Sistemas de Informação**

## **Banco de Dados Aula 05 SQL - DDL**

Dr. Diego Buchinger  
diego.buchinger@udesc.br

# Introdução

---



## SQL

### Structured Query Language

---



# Introdução

---

- SQL é uma linguagem comercial para BD relacionais
  - Desenvolvida pela IBM no início dos anos 70
  - Tornou-se padrão ISO desde a década de 80
    - ❖ SQL-1 (1986)
    - ❖ SQL-2 (1992)
    - ❖ SQL-3 (1999)
- Utiliza uma base formal combinando álgebra relacional e cálculo relacional
- Se estabeleceu como a linguagem padrão para banco de dados relacionais

# Introdução

---

- Apesar de ser chamada de “linguagem de consulta” possui outras funcionalidades também:
  - **Linguagem de Definição de Dados (DDL)**
    - Comandos para definição, remoção e modificação de tabelas, índices, chaves, restrições *etc.*
  - **Linguagem de Manipulação de Dados (DML)**
    - Comandos para consulta, inserção, exclusão e modificação de dados
  - **DML Embutida**
    - Projetada para utilização em linguagens de programação de uso geral (Cobol, C, JAVA *etc.*)
  - **Definição de Esquemas e Visões**
    - Comandos para definir esquemas e visões

# Introdução

---

- Apesar de ser chamada de “linguagem de consulta” possui outras funcionalidades também:
  - **Autenticação**
    - Comandos para especificação de autorização de acesso aos esquemas, tabelas e visões
  - **Integridade**
    - Comandos para especificar regras de integridade nas tabelas
  - **Controle de Transações**
    - Comandos para especificação de início e fim de transações, e bloqueios de dados para controle de concorrência

# Introdução

---

- A maioria dos BDs utiliza uma porta de conexão TCP para realizar a comunicação com seus serviços
- As portas podem ser definidas pelo administrador do sistema, mas utilizam um valor padrão
  - MySQL e MariaDB: 3306
  - PostgreSQL: 5432
  - Microsoft SQL Server: 1433
  - Oracle DB: 1521
  - MongoDB: 27017

Lembrando que não é interessante deixar estas portas abertas para a internet.

Pode-se deixar aberto apenas para uma rede local ou para alguns servidores de aplicação específicos.

# Qual tecnologia vamos usar?

- Vamos utilizar o **MySQL Workbench**
  - Popularidade e Comunidade: É um dos BDs mais populares, com muita documentação
  - Desempenho e Escalabilidade: É conhecido pelo seu bom desempenho e possibilidade de escalabilidade
  - Compatibilidade: Compatível com diversos SOs
  - Open Source: Software de código aberto e gratuito
  - [Já era usado e está instalado!]

**SQL e MySQL não são a mesma coisa!**

O MySQL é um SGBD e também uma variante (dialeto) do SQL e é utilizada para interagir com definições e dados dentro do BDs.



# MySQL - Instalação

---

- Para instalar, acesse a página do MySQL e baixe a versão Community adequada para seu SO.

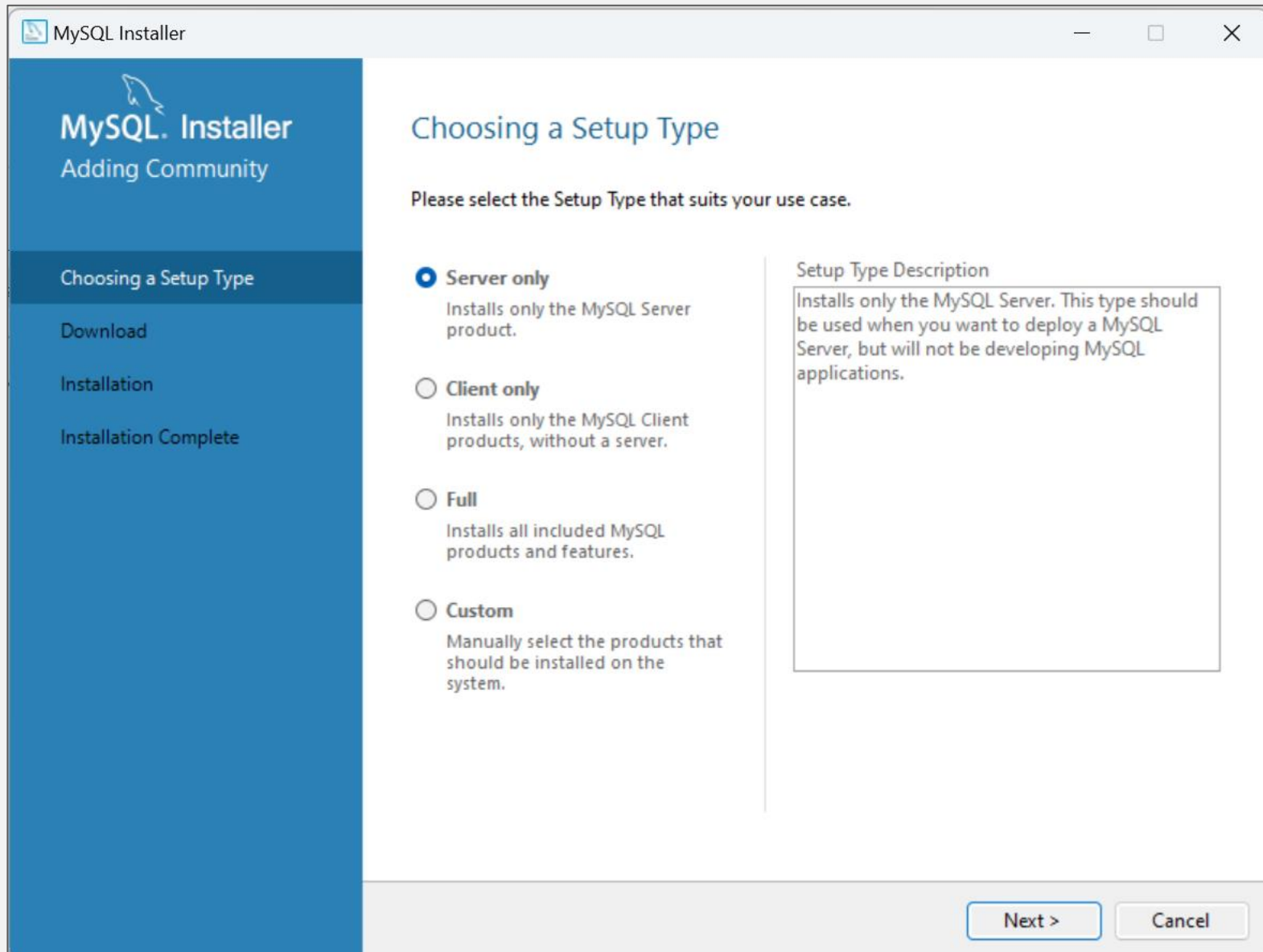
<https://dev.mysql.com/downloads/>

(precisa criar uma conta Oracle Web)

- A seguir veremos as etapas do instalador com GUI no Windows, mas as informações solicitadas devem ser as mesmas ou semelhantes para outros sistemas



# MySQL – Instalação – Server



# MySQL – Instalação – *Client*

☐ **Server only**

Installs only the MySQL Server product.

☒ **Client only**

Installs only the MySQL Client products, without a server.

☐ **Full**

Installs all included MySQL products and features.

☐ **Custom**

Manually select the products that should be installed on the system.

## Setup Type Description

Installs the tools required for MySQL application administration, but does not include the MySQL Server itself. This is useful if you intend to access and configure the server.

This Setup Type includes:

\* MySQL Shell

The new MySQL client application to manage MySQL Servers and InnoDB cluster instances.

\* MySQL Router

High availability router daemon for InnoDB cluster setups to be installed on application nodes.

\* MySQL Workbench

The GUI application to develop for and manage the server.

# MySQL – Instalação – *Full*

☐ **Server only**

Installs only the MySQL Server product.

☐ **Client only**

Installs only the MySQL Client products, without a server.

☒ **Full**

Installs all included MySQL products and features.

☐ **Custom**

Manually select the products that should be installed on the system.

## Setup Type Description

Installs all of the products available in this catalog including MySQL Server, MySQL Shell, MySQL Router, MySQL Workbench, documentation, samples and examples and more.

# MySQL – Instalação – *Custom*

## Choosing a Setup Type

Select Products

Download

Installation

Installation Complete







- ☐ **Server only**  
Installs only the MySQL Server product.
- ☐ **Client only**  
Installs only the MySQL Client products, without a server.
- ☐ **Full**  
Installs all included MySQL products and features.
- ☒ **Custom**  
Manually select the products that should be installed on the system.

## Setup Type Description

Allows you to select exactly which products you would like to install. This also allows to pick other server versions and architectures (depending on your OS).

# MySQL - Instalação – *Full*

The following products will be installed.

Product	Arch	Status	Progress	Notes
 MySQL Server 8.0.39	X64	Ready to Install		
 MySQL Workbench 8.0.38	X64	Ready to Install		
 MySQL Shell 8.0.38	X64	Ready to Install		
 MySQL Router 8.0.39	X64	Ready to Install		
 MySQL Documentation 8.0.39	X86	Ready to Install		
 Samples and Examples 8.0.39	X86	Ready to Install		

Este Computador > Disco Local (C:) > Arquivos de Programas > MySQL >			
Classificar Visualizar ...			
Nome	Data de modificação	Tipo	Tamanho
MySQL Router 8.0	04/09/2024 23:26	Pasta de arquivos	
MySQL Server 8.0	04/09/2024 23:24	Pasta de arquivos	
MySQL Shell 8.0	04/09/2024 23:26	Pasta de arquivos	
MySQL Workbench 8.0	04/09/2024 23:24	Pasta de arquivos	

# MySQL - Instalação – Configurando

## Product Configuration

We'll now walk through a configuration wizard for each of the following products.

You can cancel at any point if you wish to leave this wizard without configuring all the products.

Product	Status
MySQL Server 8.0.39	Ready to configure
MySQL Router 8.0.39	Ready to configure
Samples and Examples 8.0.39	Ready to configure

# MySQL - Instalação – Configurando

## Type and Networking

### Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type:  ▾

### Connectivity

Use the following controls to select how you would like to connect to this server.

☒ TCP/IP Port:  X Protocol Port:

☒ Open Windows Firewall ports for network access

☐ Named Pipe Pipe Name:

☐ Shared Memory Memory Name:

### Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

☐ Show Advanced and Logging Options



# MySQL - Instalação – Configurando

## Type and Networking

### Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: Development Computer

### Connectivity

Use the following

☒ TCP/IP

☒ ODBC

☐ Named Pipes

☐ Shared

### Advanced Configuration

Select the checkboxes for advanced configuration and logging options.

☐ Show Advanced and Logging Options



#### Development Computer

This is a development computer, and many other applications will be installed on it. A minimal amount of memory will be used by MySQL.



#### Server Computer

Several server applications will be running on this computer. Choose this option for web or application servers. MySQL will have medium memory usage.



#### Dedicated Computer

This computer is dedicated to running the MySQL database server. No other servers, such as web servers, will be run. MySQL will make use of all available memory.



# MySQL - Instalação – Configurando

## Authentication Method

### ☒ Use Strong Password Encryption for Authentication (RECOMMENDED)

MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward.



Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching\_sha2\_password authentication).

Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.

### ☐ Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)

Using the old MySQL 5.x legacy authentication method should only be considered in the following cases:

- If applications cannot be updated to use MySQL 8 enabled Connectors and drivers.
- For cases where re-compilation of an existing application is not feasible.
- An updated, language specific connector or driver is not yet available.

Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.

# MySQL - Instalação – Configurando

## Accounts and Roles

### Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

●●●●●●●●

Repeat Password:

●●●●●●●●

Password strength: **Weak**

### MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

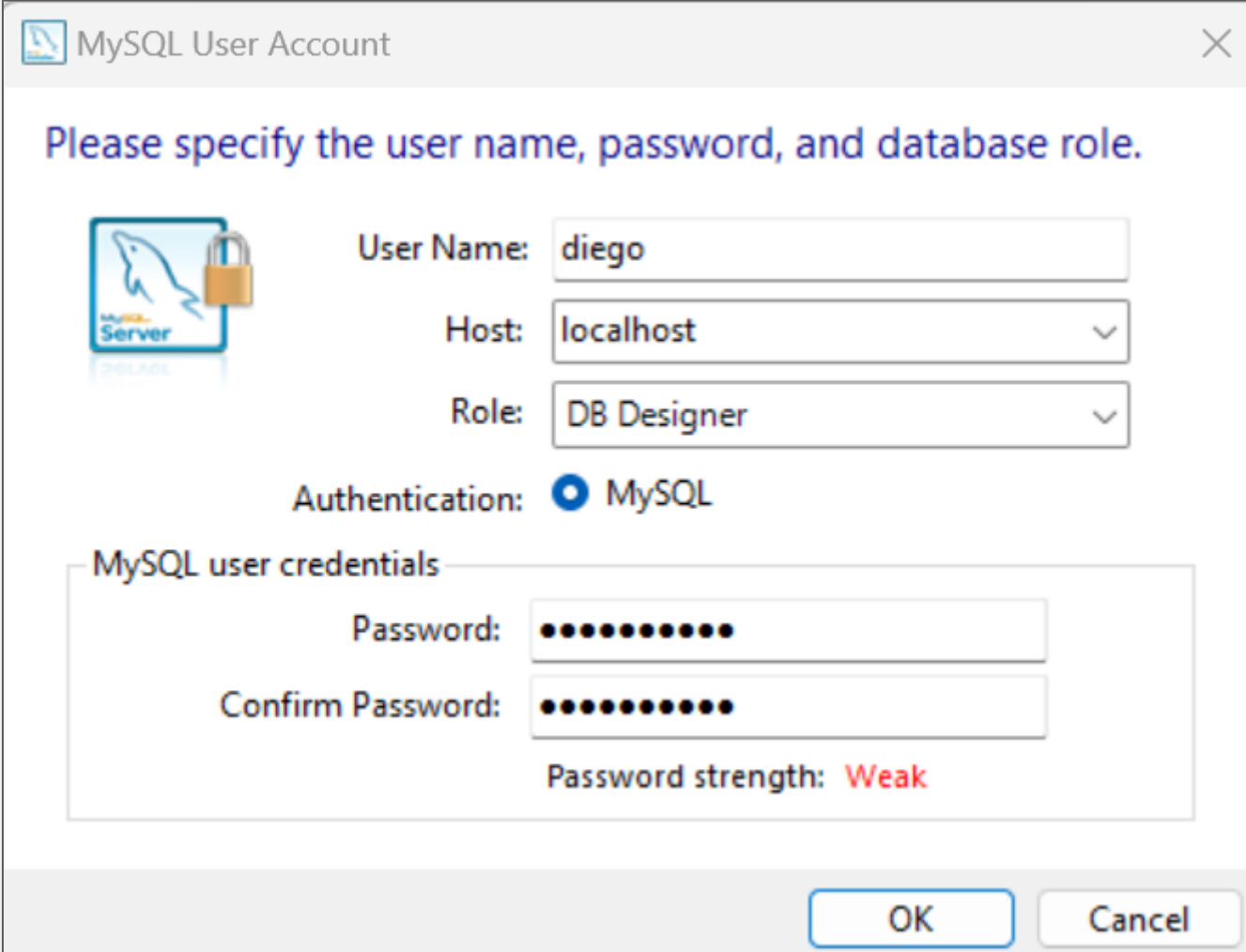
MySQL User Name	Host	User Role	

Add User

Edit User

Delete


# MySQL - Instalação – Configurando



The image shows a 'MySQL User Account' dialog box. It has a title bar with the MySQL logo and a close button. The main text says 'Please specify the user name, password, and database role.' On the left is a MySQL Server icon with a padlock. The form contains fields for 'User Name' (diego), 'Host' (localhost), and 'Role' (DB Designer). There is a radio button for 'Authentication' set to 'MySQL'. Below these is a section titled 'MySQL user credentials' containing 'Password' and 'Confirm Password' fields, both masked with dots. A 'Password strength' indicator shows 'Weak' in red. At the bottom are 'OK' and 'Cancel' buttons.

MySQL User Account

Please specify the user name, password, and database role.

 User Name:

Host:

Role:

Authentication: ☒ MySQL

MySQL user credentials

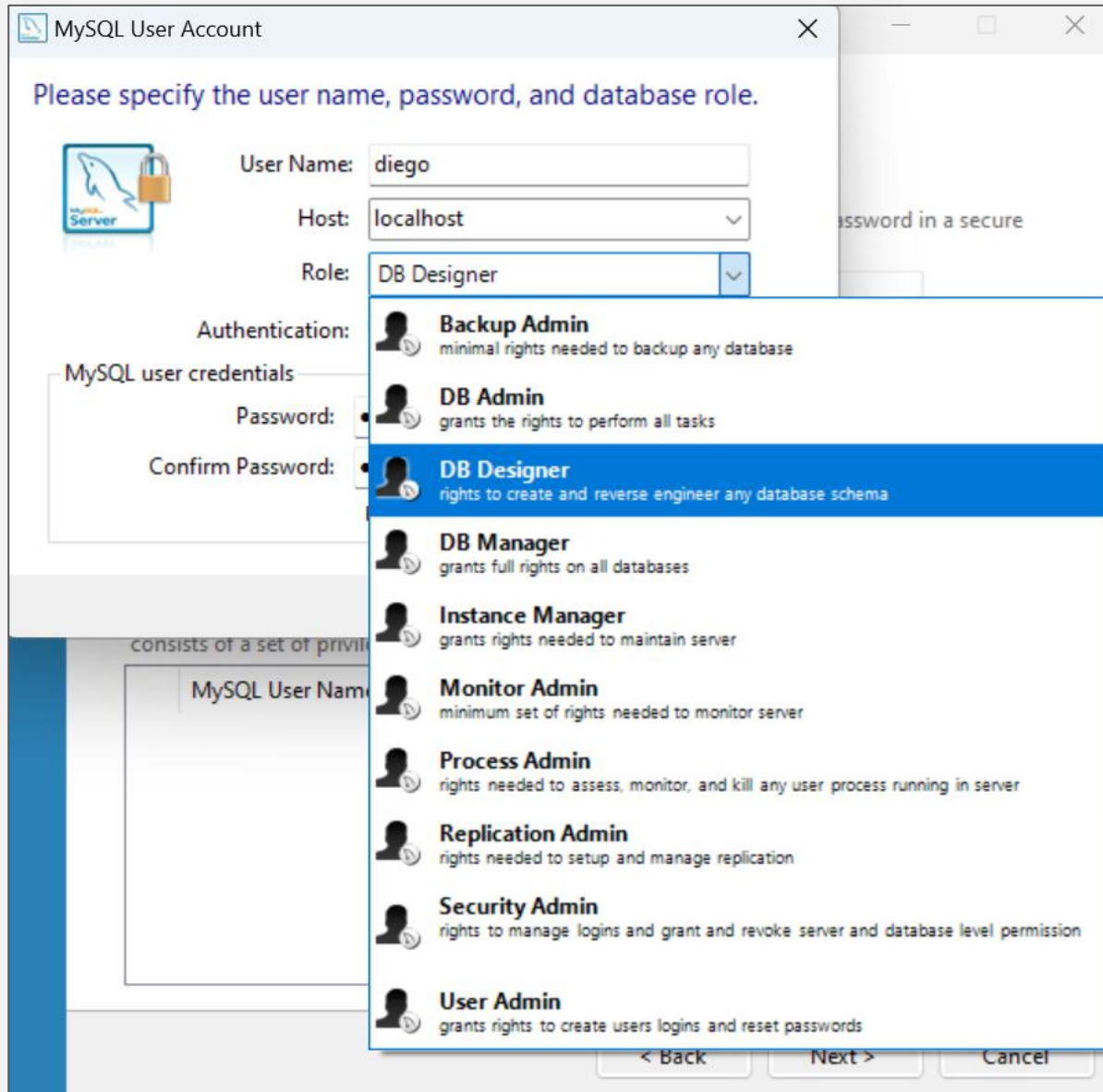
Password:

Confirm Password:

Password strength: **Weak**

OK Cancel

# MySQL - Instalação – Configurando



The image shows the 'MySQL User Account' configuration window. It prompts the user to specify the user name, password, and database role. The 'User Name' field is set to 'diego', the 'Host' is 'localhost', and the 'Role' is 'DB Designer'. A list of roles is displayed, with 'DB Designer' selected. The roles include Backup Admin, DB Admin, DB Designer, DB Manager, Instance Manager, Monitor Admin, Process Admin, Replication Admin, Security Admin, and User Admin, each with a brief description of its permissions. The 'Authentication' section is partially visible, showing fields for 'MySQL user credentials', 'Password', and 'Confirm Password'.

MySQL User Account

Please specify the user name, password, and database role.

User Name: diego

Host: localhost

Role: DB Designer

Authentication:

MySQL user credentials

Password:

Confirm Password:

MySQL User Name

consists of a set of privi

MySQL User Name

Backup Admin  
minimal rights needed to backup any database

DB Admin  
grants the rights to perform all tasks

DB Designer  
rights to create and reverse engineer any database schema

DB Manager  
grants full rights on all databases

Instance Manager  
grants rights needed to maintain server

Monitor Admin  
minimum set of rights needed to monitor server

Process Admin  
rights needed to assess, monitor, and kill any user process running in server

Replication Admin  
rights needed to setup and manage replication

Security Admin  
rights to manage logins and grant and revoke server and database level permission

User Admin  
grants rights to create users logins and reset passwords

< Back Next > Cancel

# MySQL - Instalação – Configurando

## Windows Service

☒ Configure MySQL Server as a Windows Service

### Windows Service Details

Please specify a Windows Service name to be used for this MySQL Server instance. A unique name is required for each instance.

Windows Service Name:

☒ Start the MySQL Server at System Startup

### Run Windows Service as ...

The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

☒ Standard System Account

Recommended for most scenarios.

☐ Custom User

An existing user account can be selected for advanced scenarios.

# MySQL - Instalação – Configurando

## Server File Permissions

MySQL Installer can secure the server's data directory by updating the permissions of files and folders located at:

C:\ProgramData\MySQL\MySQL Server 8.0\Data

Do you want MySQL Installer to update the server file permissions for you?

- ☒ Yes, grant full access to the user running the Windows Service (if applicable) and the administrators group only. Other users and groups will not have access.
- ☐ Yes, but let me review and configure the level of access.
- ☐ No, I will manage the permissions after the server configuration.

# MySQL - Instalação – Configurando

## Apply Configuration

The configuration operation has completed.


Configuration Steps Log

- ✓ Writing configuration file
- ✓ Updating Windows Firewall rules
- ✓ Adjusting Windows service
- ✓ Initializing database (may take a long time)
- ✓ Updating permissions for the data folder and related server files
- ✓ Starting the server
- ✓ Applying security settings
- ✓ Creating user accounts
- ✓ Updating the Start menu link

The configuration for MySQL Server 8.0.39 was successful.  
Click Finish to continue.



# MySQL - Instalação – Configurando



## MySQL. Installer

MySQL Router 8.0.39

### MySQL Router Configuration

### MySQL Router Configuration

☐ Bootstrap MySQL Router for use with InnoDB Cluster

This wizard can bootstrap MySQL Router to direct traffic between MySQL applications and InnoDB Cluster. Applications that connect to the router will be automatically directed to an available read/write or read-only member of the cluster.

The bootstrapping process requires a connection to InnoDB Cluster. In order to register the MySQL Router for monitoring, use the current Read/Write instance of the cluster.

Hostname:

Port:

Management User:

Password:

MySQL Router requires specification of a base port (between 80 and 65532). The first port is used for classic read/write connections. The other ports are computed sequentially after the first port. If any port is indicated to be in use, please change the base port.

Classic MySQL protocol connections to InnoDB Cluster:

Read/Write:

Read Only:


X Protocol connections to InnoDB Cluster:

Read/Write:

Read Only:



# MySQL - Instalação – Configurando

**MySQL. Installer**  
Samples and Examples

Connect To Server

Apply Configuration

## Connect To Server

Select the MySQL server instances from the list to receive sample schemas and data.

	Server	Port	Arch...	Type	Status
<input checked="" type="checkbox"/>	MySQL Server 8.0.39	3306	X64	Stand-alone Server	Running

Provide the credentials that should be used (requires root privileges).  
Click "Check" to ensure they work.

User name:  Credentials provided in Server configuration

Password:

# MySQL - Instalação – Configurando

## Apply Configuration

The configuration operation has completed.

Configuration Steps

Log

✓

 Checking if there are any features installed that need configuration.

✓

 Running Scripts

The configuration for Samples and Examples 8.0.39 was successful.  
Click Finish to continue.

# MySQL - Instalação – *The End*

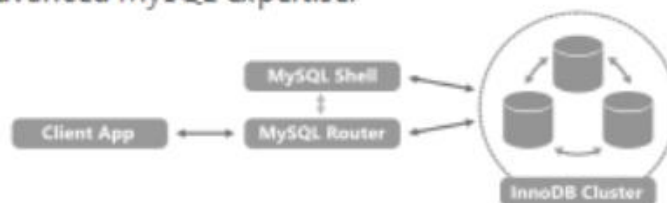
## Installation Complete

The installation procedure has been completed.

Copy Log to Clipboard

- ☒ Start MySQL Workbench after setup
- ☒ Start MySQL Shell after setup

The MySQL Shell is an advanced MySQL client application that can be used to work with single MySQL Server instances. Further, it can be used to create and manage InnoDB Cluster, an integrated solution for high availability and scalability of MySQL databases, without requiring advanced MySQL expertise.



Refer to the following links for documentation, tutorials and examples on MySQL Shell:

[MySQL Shell Documentation](#)

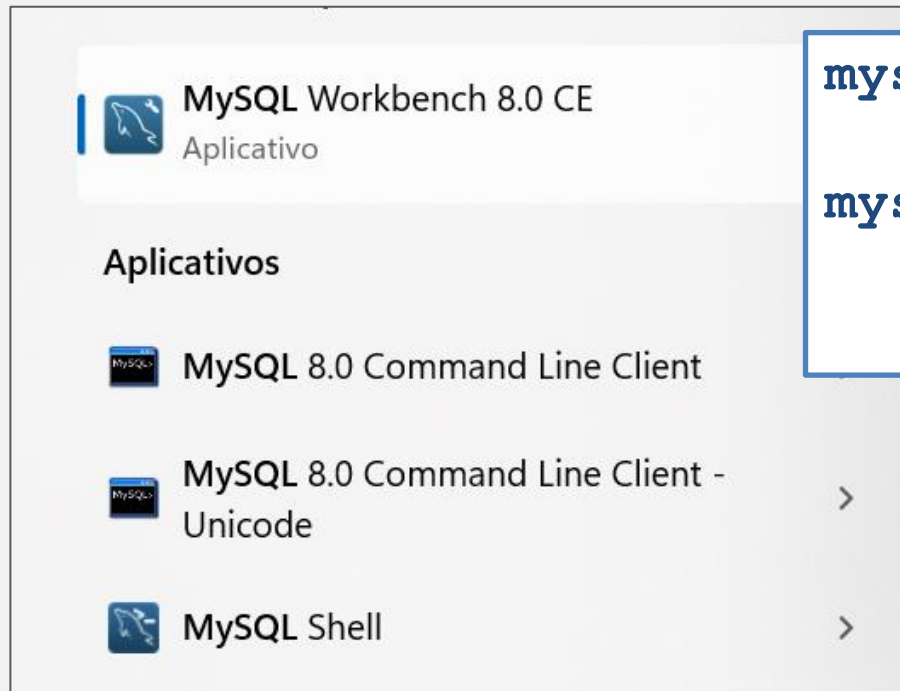
[Setting up a Real World Cluster Blog](#)

[The All New MySQL InnoDB ReplicaSet Blog](#)

[Changing Cluster Options Live Blog](#)

# MySQL – Conectando

- Algumas ferramentas podem ser utilizadas para se conectar ao MySQL
- Vamos iniciar pela linha de comando



```
mysql -u USUARIO -p
```

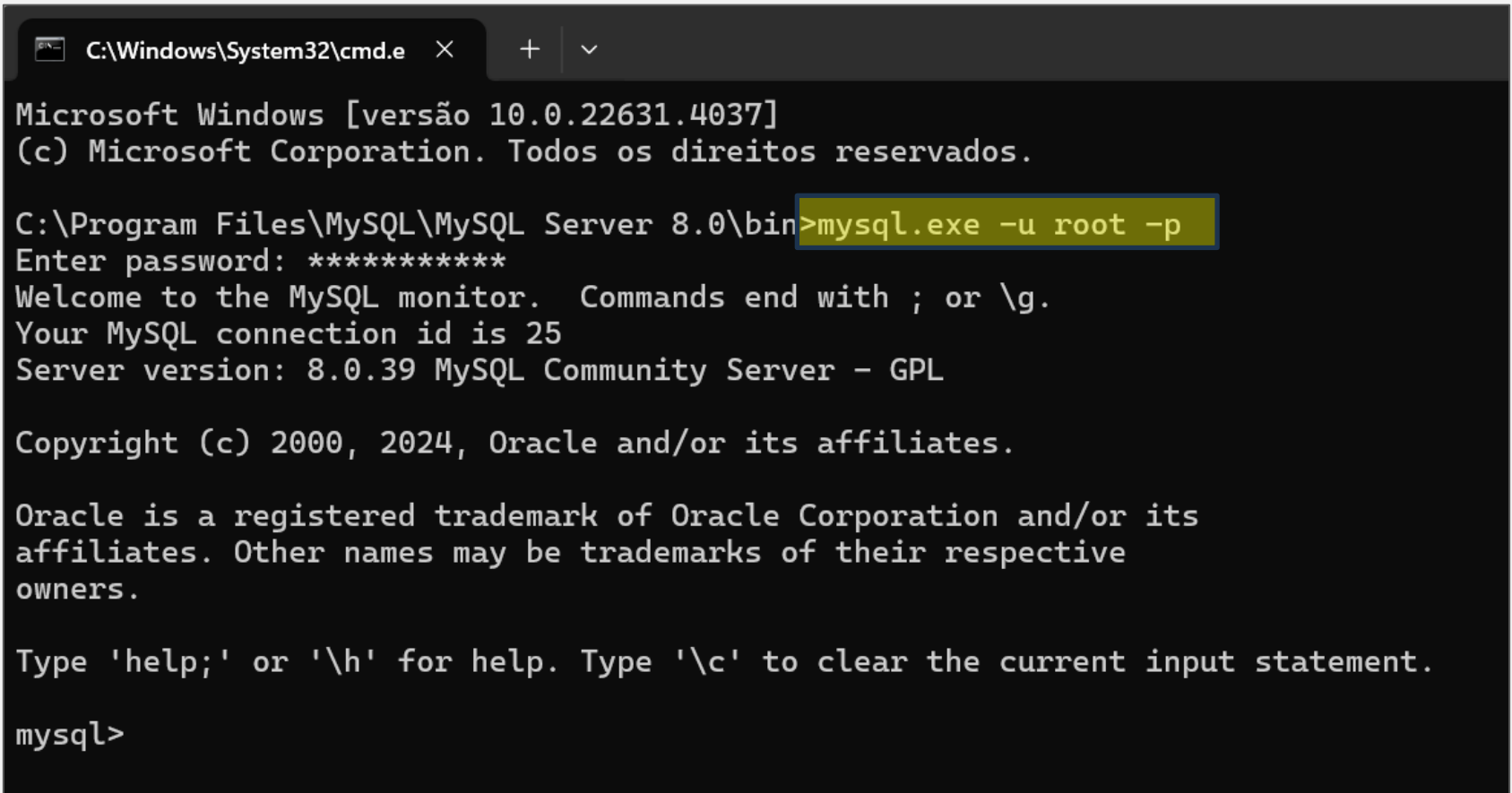
```
mysql -u USUARIO  
[-p senha]  
[-h HOSTNAME DATABASE]
```

[...] = opcional

Nos labs existem duas  
versões instaladas:  
(1) Padrão do MySQL  
(2) Versão do XAMPP  
(e as portas?)  
A senha é: udesc

# MySQL – Base de Dados

- Terminal ou Console



```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [versão 10.0.22631.4037]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql.exe -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

# MySQL – Base de Dados

- Terminal ou Console

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u diego -pUdescDiego -h localhost world
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

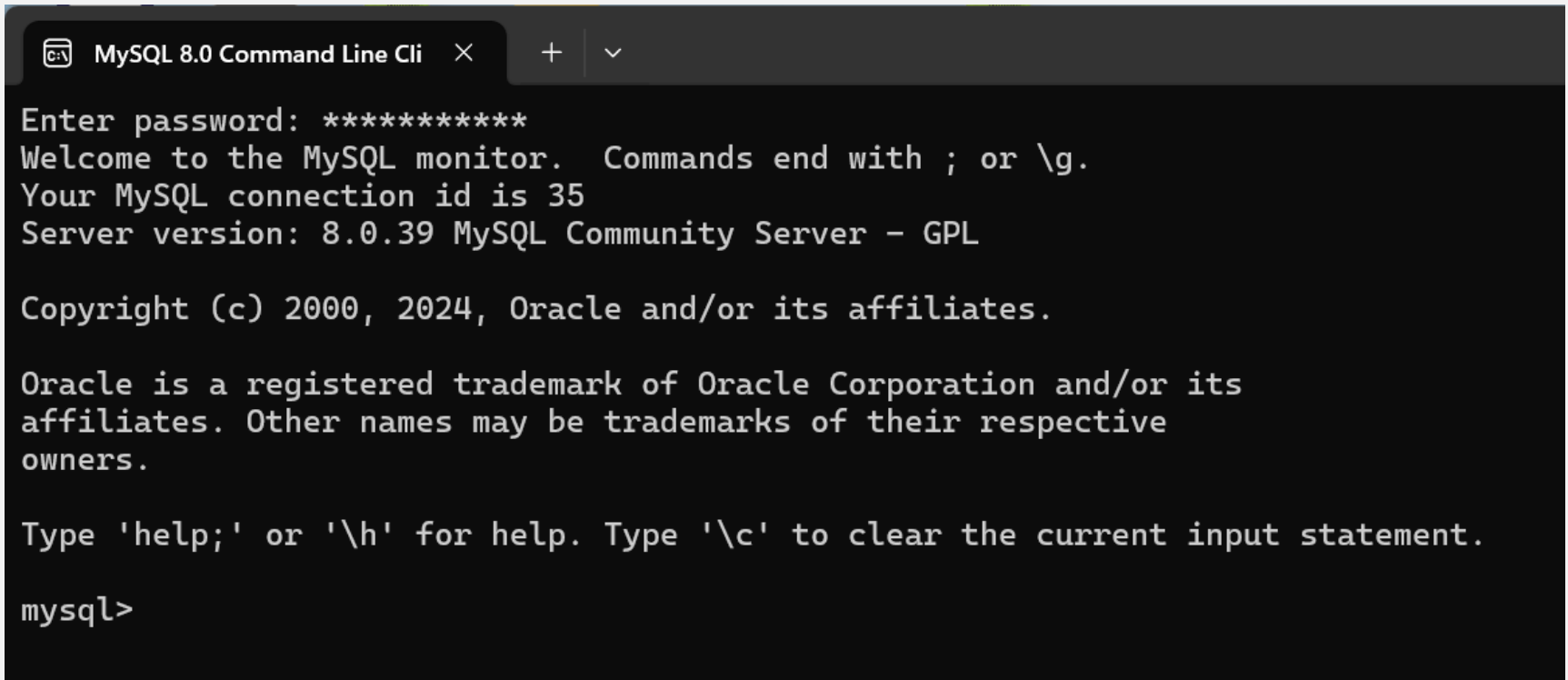
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Como verificar/confirmar com qual  
usuário estou conectado?

```
SELECT USER();
```

# MySQL – Base de Dados

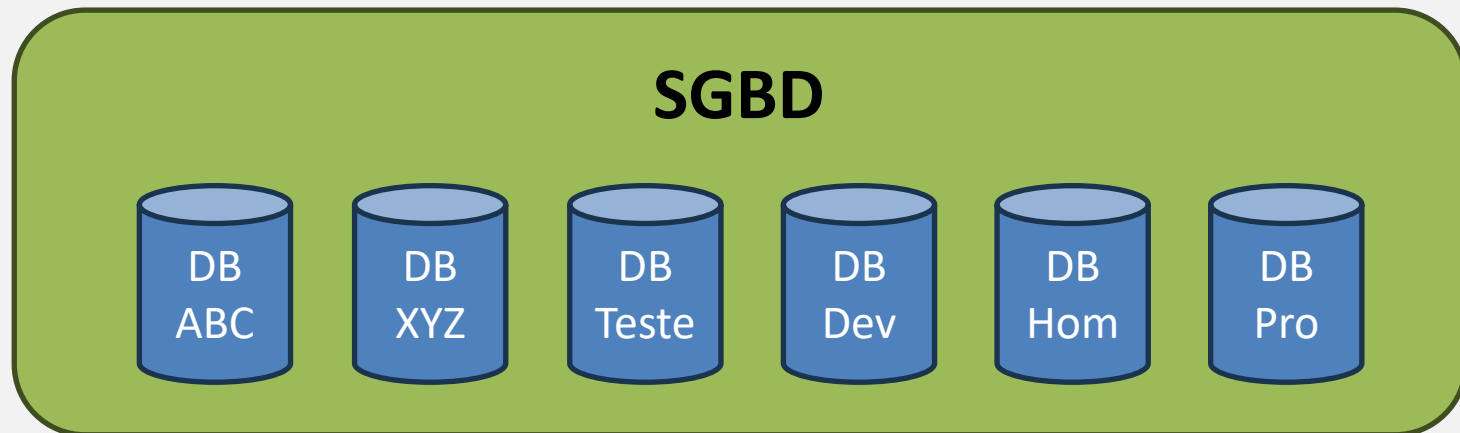
- Terminal do MySQL
  - Conecta como root



```
MySQL 8.0 Command Line Cli  ×  +  ▾  
Enter password: *****  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 35  
Server version: 8.0.39 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql>
```

# MySQL – Banco de Dados

- Os bancos de dados são containers que armazenam e organizam os dados e suas estruturas
  - BDs contém: tabelas, visões, procedures e funções
  - Cada BD é independente e isolado dos demais





# MySQL – Banco de Dados

```
SHOW DATABASES;
```

Mostra as bases de dados

```
CREATE DATABASE exemplo;
```

Cria uma nova base de dados

```
DROP DATABASE exemplo;
```

Remove uma base de dados

```
USE banco_dados;
```

Seleciona uma base de dados

```
SELECT DATABASE();
```

Verifica qual base de dados está selecionada (em uso)

```
quit
```

Finaliza a conexão com o SGBD

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.00 sec)
```

```
mysql> use world;
Database changed
mysql>
```

Se o usuário não tiver privilégios,  
um erro como este deve aparecer:  
ERROR 1044 (42000): Access denied for  
user 'xyz'@'localhost' to database  
'exemplo'

```
mysql> select database();
+-----+
| database() |
+-----+
| world |
+-----+
1 row in set (0.00 sec)

mysql>
```

# MySQL – Esquemas

- No MySQL, esquemas (schema) e bases de dados (*databases*) podem ser considerados como sinônimos
- Em diversos outros SGBDs, as bases de dados tem um ou mais esquemas, onde estes servem como/para:
  - Uma área organizada dentro do BD, como um container, para armazenar objetos como tabelas, visões, índices *etc.*
  - Atribuir a propriedade de espaço para um usuário
  - Permitir melhor organização, segurança e modularidade
  - Ajudar a evitar conflito entre nomes de objetos de diferentes usuários ou aplicações

```
CREATE SCHEMA nome_esquema;
```

Cria um novo esquema

```
DROP SCHEMA nome_esquema;
```

Remove um esquema

# MySQL – Criando usuários

- Os bancos de dados tem maior segurança com o gerenciamento adequado de usuários e suas permissões

## **CREATE USER**

```
'nome_usuario'@'host'  
[IDENTIFIED BY 'senha'];
```

Cria um novo usuário especificando o host de onde ele pode se conectar (ex: 'localhost', '%')

OBS: se uma senha não for definida para um usuário com host '%', ele não terá acesso ao SGBD! Neste caso é necessário definir uma senha com o comando:

```
SET PASSWORD FOR 'nome_usuario'@'host' = 'nova_senha';
```

```
DROP USER 'nome_usuario'@'host';
```

Remove o usuário especificado

**Ex:**

```
CREATE USER 'teste'@'localhost'  
IDENTIFIED BY 'senha';
```

# MySQL – Gerenciando permissões

- Os bancos de dados tem maior segurança com o gerenciamento adequado de usuários e suas permissões

```
GRANT privilegio  
ON banco_dados.tabela  
TO 'nome_usuario'@'host';
```

Concede um privilégio em uma (ou todas) tabela de um BD específico a um determinado usuário

- privilégios:** SELECT, INSERT, UPDATE, DELETE, ALL PRIVILEGES
- banco\_dados.tabela especifica a tabela que terá a permissão.  
Para especificar todas as tabelas de um BD, utilize: banco\_dados.\*

**Ex:**

```
GRANT SELECT ON world.*  
TO 'teste'@'localhost';
```

Ao tentar executar uma ação sem ter as devidas permissões, o banco acusa um erro. Ex:

```
ERROR 1142 (42000): INSERT command denied to user  
'teste'@'localhost' for table 'city'
```

# MySQL – Gerenciando permissões

- Os bancos de dados tem maior segurança com o gerenciamento adequado de usuários e suas permissões

```
REVOKE privilegio  
ON banco_dados.tabela  
TO 'nome_usuario'@'host';
```

Revoga um privilégio em uma (ou todas) tabela de um BD específico a um determinado usuário

- privilégios:** SELECT, INSERT, UPDATE, DELETE, ALL PRIVILEGES
- banco\_dados.tabela especifica a tabela que terá a permissão.  
Para especificar todas as tabelas de um BD, utilize: banco\_dados.\*

```
SHOW GRANTS  
FOR 'nome_usuario'@'host';
```

Mostra os privilégios de um determinado usuário

O privilégio **USAGE** é uma permissão mínima para se conectar a uma base de dados, mas sem conceder privilégio sobre os objetos (ex: usuário de monitoramento).

# MySQL – Info das Tabelas

- Existem diversos comandos que podem ser utilizados para verificar informações das tabelas

```
SHOW TABLES;
```

Cria um novo esquema

```
SHOW CREATE TABLE tabela;
```

Mostra o comando de criação da tabela especificada

```
DESCRIBE tabela;
```

Mostra os principais dados de cabeçalho da tabela

```
SHOW INDEXES FROM tabela;  
SHOW KEYS FROM tabela;
```

Exibe informações específicas de uma tabela (ex: índices e chaves)

```
SHOW TABLE STATUS FROM base_dados;
```

Mostra informações de status das tabelas de uma base de dados específica

# MySQL – Info das Tabelas

```
-----  
| city | CREATE TABLE `city` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `Name` char(35) NOT NULL DEFAULT '',  
  `CountryCode` char(3) NOT NULL DEFAULT '',  
  `District` char(20) NOT NULL DEFAULT '',  
  `Population` int NOT NULL DEFAULT '0',  
  PRIMARY KEY (`ID`),  
  KEY `CountryCode` (`CountryCode`),  
  CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)  
) ENGINE=InnoDB AUTO_INCREMENT=5001 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |  
-----+
```

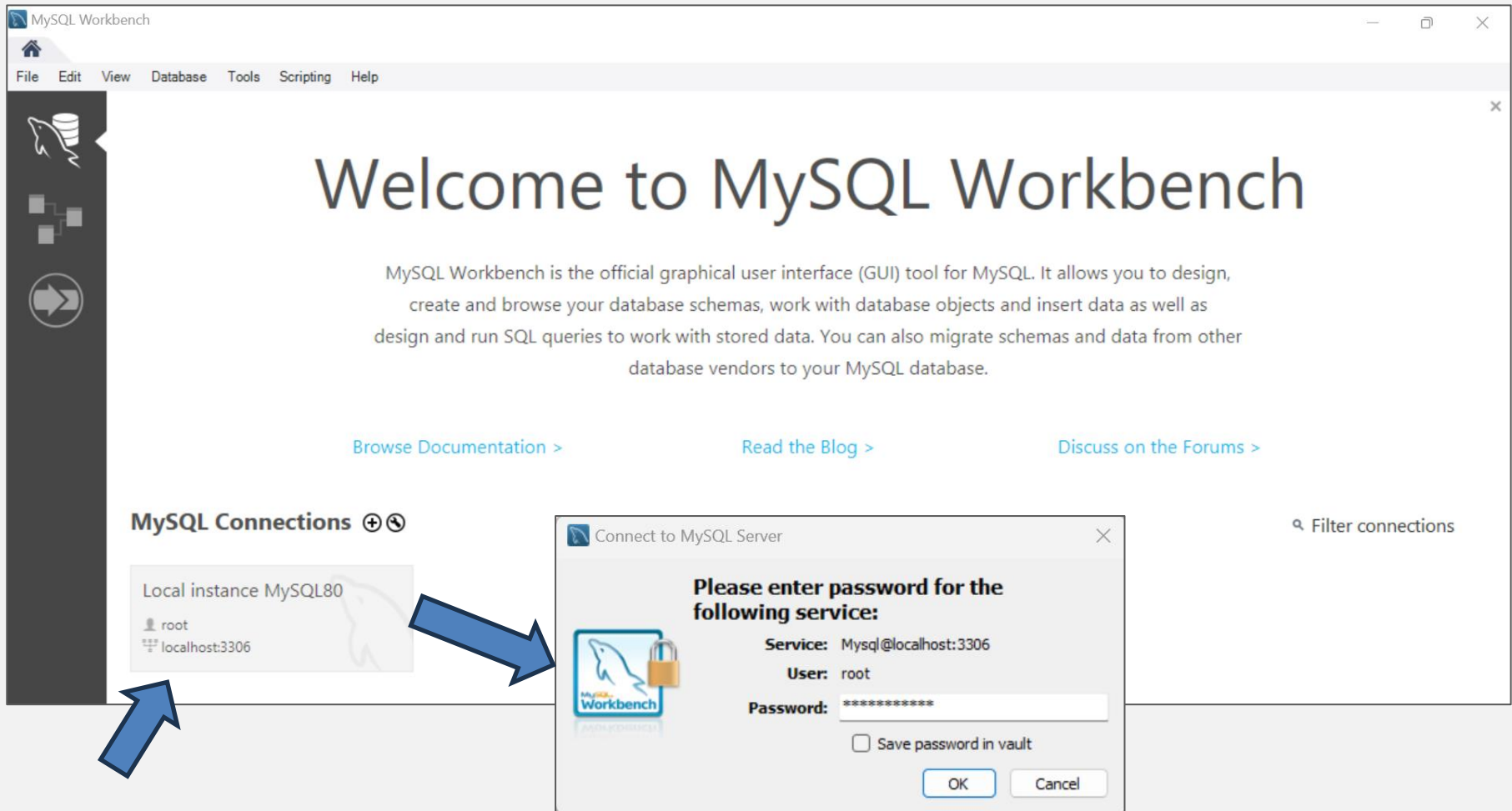
```
mysql> DESCRIBE city;
```

Field	Type	Null	Key	Default	Extra
ID	int	NO	PRI	NULL	auto_increment
Name	char(35)	NO			
CountryCode	char(3)	NO	MUL		
District	char(20)	NO			
Population	int	NO		0	

```
5 rows in set (0.00 sec)
```

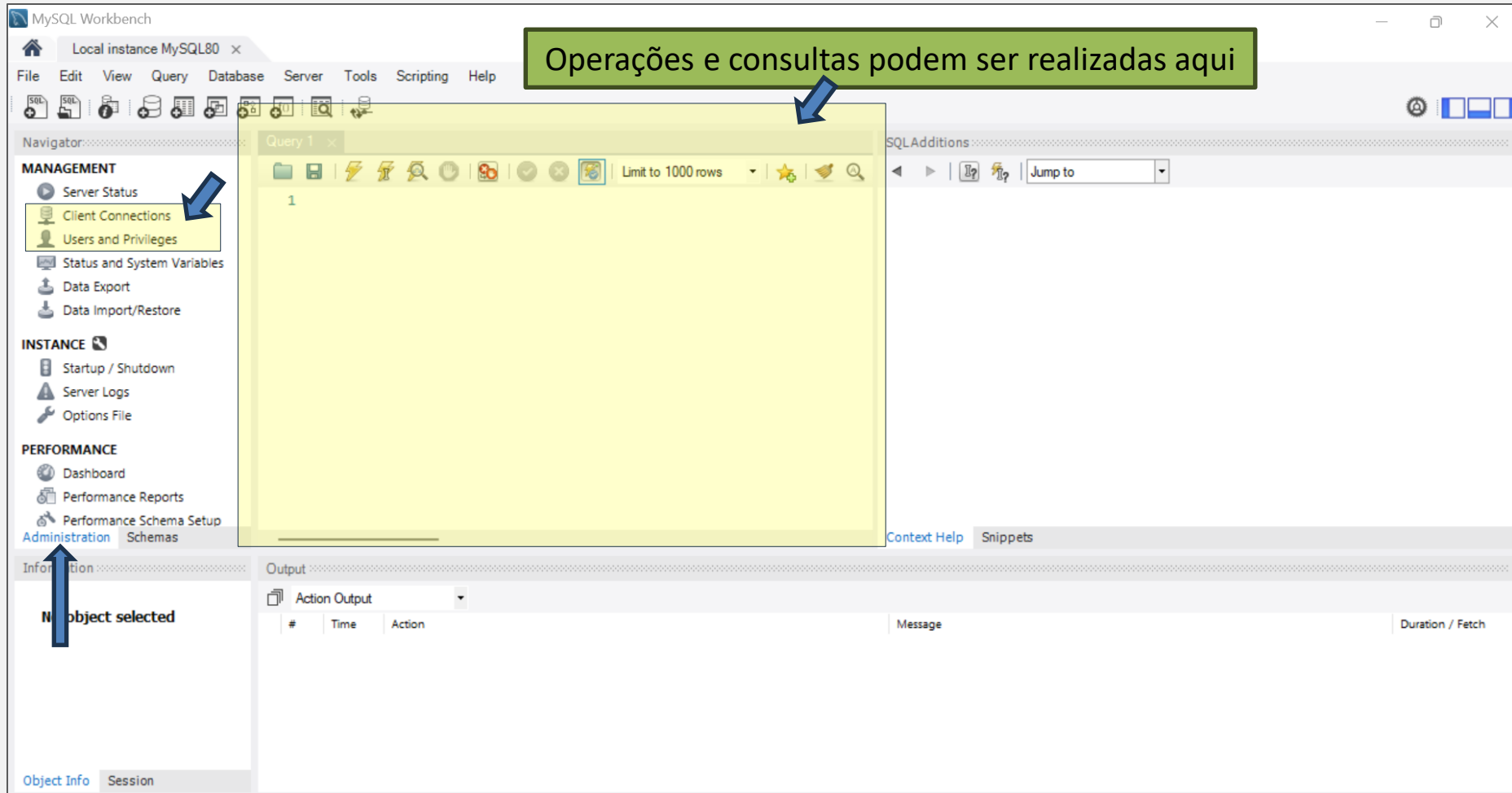
# MySQL – Conectando

- Agora utilizando a GUI – MySQL Workbench

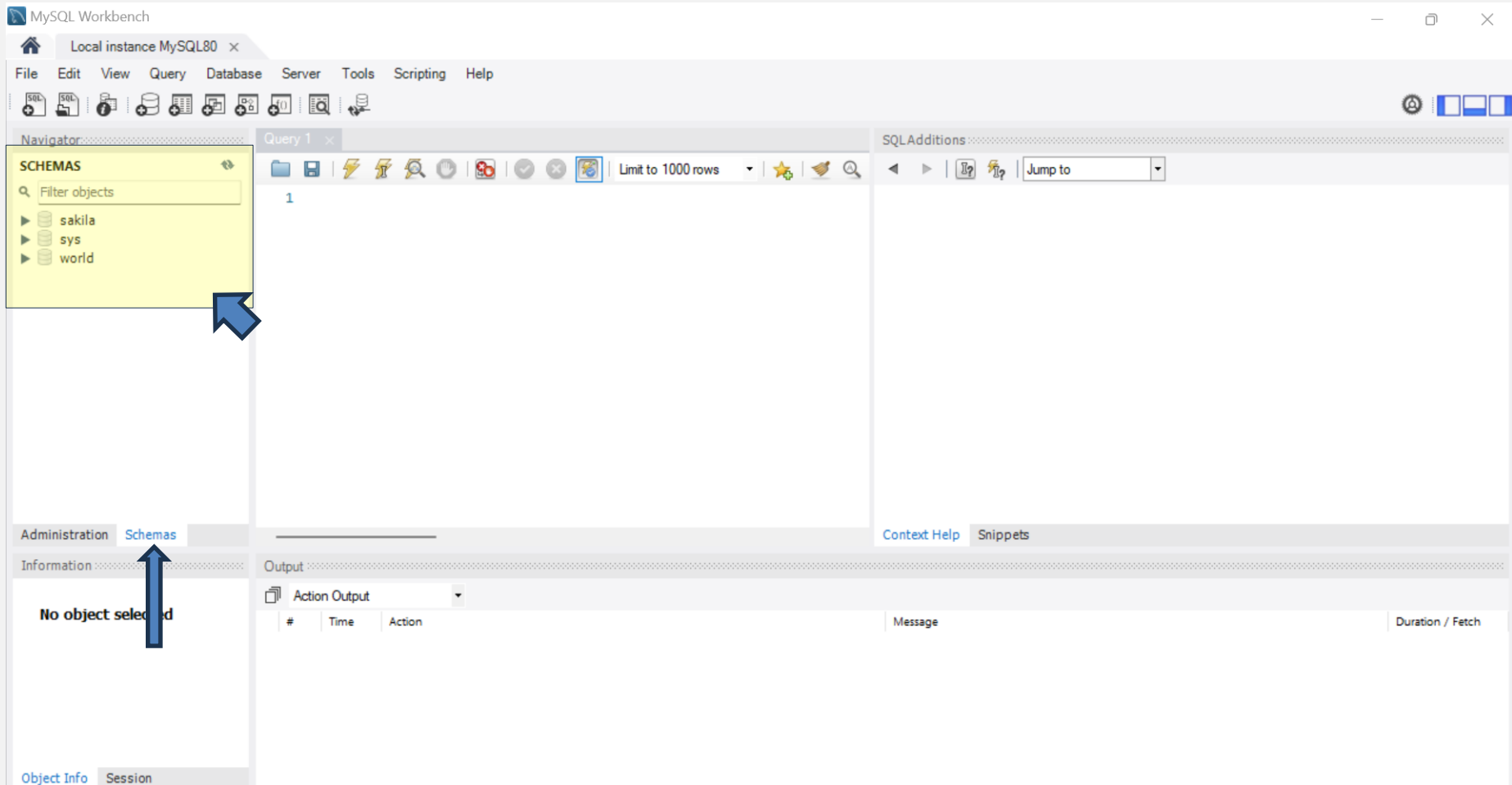




# MySQL – GUI Tour



# MySQL – GUI Tour



# MySQL – GUI Tour

- *Administration – Users and Privileges*

Local instance MySQL80  
**Users and Privileges**

User Accounts

User	From Host
diego	localhost
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

**Details for account diego@localhost**

Login Account Limits Administrative Roles Schema Privileges

Schema	Privileges
world	INSERT, SELECT

Schema and Host fields may use % and \_ wildcards.  
The server will match specific entries before wildcarded ones.

The user 'diego'@'localhost' will have the following access rights to the schema 'world':

**Object Rights**

- ☒ SELECT
- ☒ INSERT
- ☐ UPDATE
- ☐ DELETE
- ☐ EXECUTE
- ☐ SHOW VIEW

**DDL Rights**

- ☐ CREATE
- ☐ ALTER
- ☐ REFERENCES
- ☐ INDEX
- ☐ CREATE VIEW
- ☐ CREATE ROUTINE
- ☐ ALTER ROUTINE
- ☐ EVENT
- ☐ DROP
- ☐ TRIGGER

**Other Rights**

- ☐ GRANT OPTION
- ☐ CREATE TEMPORARY TABLES
- ☐ LOCK TABLES

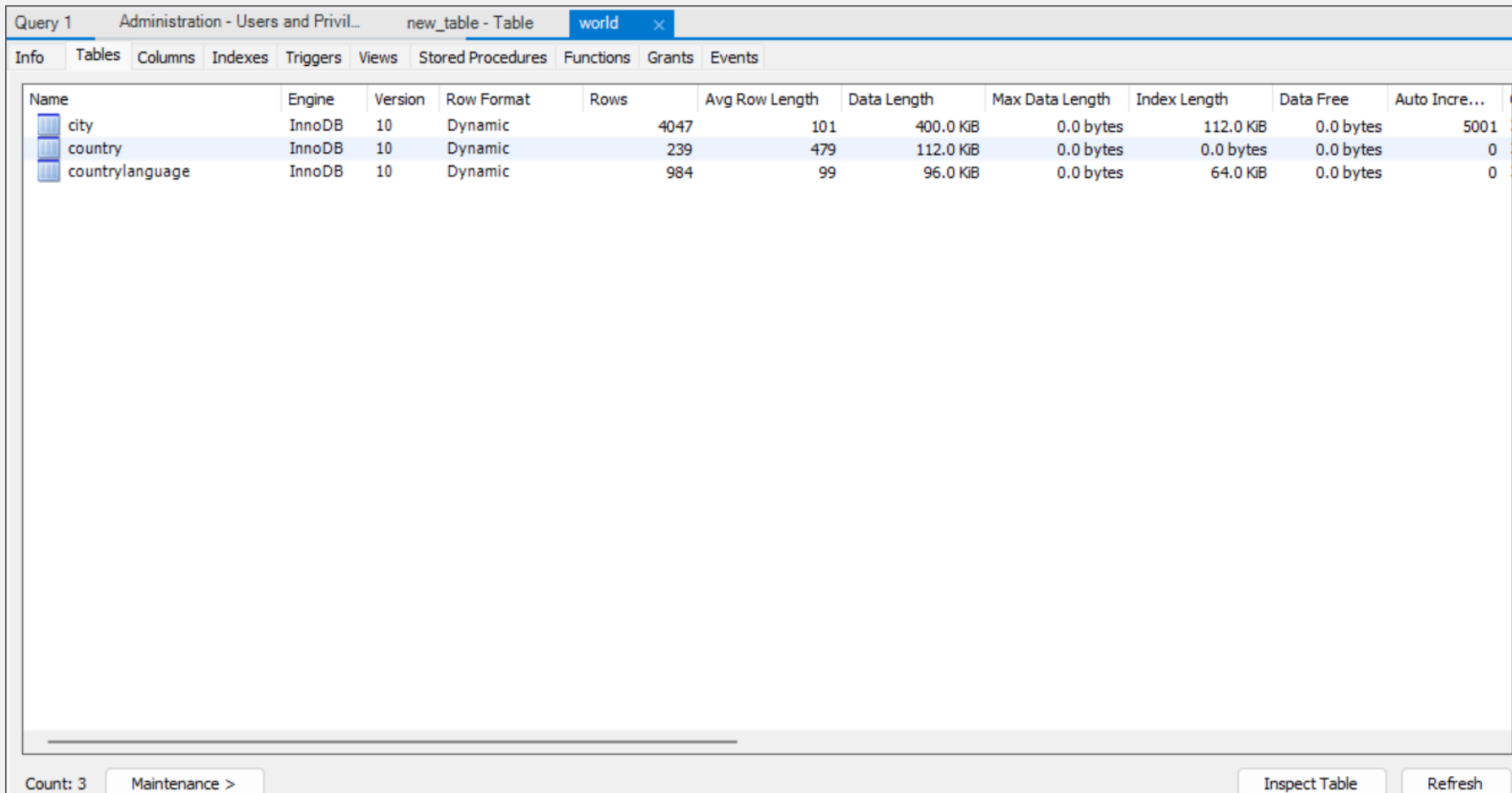
Revoke All Privileges Delete Entry Add Entry...

Unselect All Select All

Add Account Delete Refresh Revert Apply

# MySQL – GUI Tour

- *Schemas – Schema Inspector*



The screenshot shows the MySQL Schema Inspector interface. At the top, there are tabs for 'Query 1', 'Administration - Users and Privil...', 'new\_table - Table', and 'world'. Below these are sub-tabs for 'Info', 'Tables', 'Columns', 'Indexes', 'Triggers', 'Views', 'Stored Procedures', 'Functions', 'Grants', and 'Events'. The 'Tables' sub-tab is selected, displaying a table with columns: Name, Engine, Version, Row Format, Rows, Avg Row Length, Data Length, Max Data Length, Index Length, Data Free, and Auto Incre... The table lists three tables: 'city', 'country', and 'countrylanguage'. At the bottom left, it says 'Count: 3' and 'Maintenance >'. At the bottom right, there are buttons for 'Inspect Table' and 'Refresh'.

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incre...
city	InnoDB	10	Dynamic	4047	101	400.0 KiB	0.0 bytes	112.0 KiB	0.0 bytes	5001
country	InnoDB	10	Dynamic	239	479	112.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
countrylanguage	InnoDB	10	Dynamic	984	99	96.0 KiB	0.0 bytes	64.0 KiB	0.0 bytes	0

# MySQL – Criando Tabelas

- A criação de tabelas possui diversos campos opcionais

```
CREATE TABLE nome_tabela(  
    nome_atributo_1 TIPO_1 [*RESTRIÇÕES*],  
    [{nome_atributo_n TIPO_N [*RESTRIÇÕES*]}],  
    [PRIMARY KEY (nome(s)_atributo(s))],  
    [FOREIGN KEY (nome_atributo)  
        REFERENCES nome_tabela (nome_atributo)]  
);
```

{...} = conjunto de valores

**nomes\_atributos** => as restrições são semelhantes às de definição de variáveis

## RESTRIÇÕES BÁSICAS

- NOT NULL: indica que o campo não pode ser nulo, ou seja, precisa ter um valor
- PRIMARY KEY: indica que um campo é chave primária (PK);
- REFERENCES tabela (campo): indica que um campo é chave estrangeira (FK)
- UNIQUE: indica que um campo deve ser único para cada registro / linha da tabela (não pode haver duplicatas)

**OBS:** restrições PK e FK  
podem ser definidas no final

# MySQL – Criando Tabelas

```
CREATE TABLE Cursos(  
    id INT,  
    nome VARCHAR(30) NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Disciplinas(  
    sigla CHAR(3) PRIMARY KEY,  
    curso_id INT REFERENCES cursos (id),  
    ch SMALLINT  
);
```

```
CREATE TABLE Alunos(  
    matricula INT,  
    nome VARCHAR(50) NOT NULL,  
    curso_id INT,  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (curso_id) REFERENCES Cursos (id)  
);
```

# MySQL – Criando Tabelas

- **Principais tipos de dados:**
  - Numéricos: englobam números inteiros de vários tamanhos e pontos flutuantes

nome	tamanho armazenamento	faixa de valores
TINYINT	1 byte	-128 a 127
SMALLINT	2 bytes	-32.768 a +32.767
INT	4 bytes	-2.147.483.648 a +2.147.483.647
BIGINT	8 bytes	-9.223.372.036.854.775.808 a ...
DECIMAL (M, D)	Variável	M = número de dígitos D = dígitos após a vírgula
FLOAT	4 bytes	Precisão de 6 dígitos decimais
DOUBLE	8 bytes	Precisão de 15 dígitos decimais

**OBS:** tamanho exato pode variar dependendo da configuração do servidor MySQL e da plataforma

# MySQL – Criando Tabelas

- **Principais tipos de dados:**

Pode-se utilizar atributos opcionais para dados numéricos

- UNSIGNED: indica que um tipo numérico não terá sinal e por consequência a faixa de valores é alterada
- ZEROFILL: indica que os espaços vazios do campo serão preenchidos com zeros [ex: PostgreSQL não implementa]
- (n) : É possível especificar o “tamanho de apresentação”,  
ex: int(5) => 00032 (OBS: faixa de valores não muda)
- (i, d) : para o tipo decimal é possível especificar o número de dígitos totais e decimais,  
ex: decimal(6,2) => 9.999,99 até +9.999,99



# MySQL – Criando Tabelas

- **Principais tipos de dados:**
  - Cadeia de Caracteres: podem ser de tamanho fixo [ `char(n)` ] ou variável [ `varchar(n)`, `text` ]
  - Booleano: assume valores `true` e `false`, mas pode ser também um valor `null`
  - Date: formato `aaaa-mm-dd`
  - Time: formato `hh:mm:ss`
  - Timestamp: formado por data e hora
- Cada SGBD possui suas peculiaridades, podendo ter notações ou tipos de dados diferentes (específicos)
  - Por exemplo, o PostgreSQL permite o uso de dados para armazenar endereços IPv4, IPv6 e MAC

# MySQL – Criando Tabelas

- **Principais tipos de dados:**

Pode-se utilizar atributos opcionais para dados numéricos

- UNSIGNED: indica que um tipo numérico não terá sinal e por consequência a faixa de valores é alterada
- ZEROFILL: indica que os espaços vazios do campo serão preenchidos com zeros [ex: PostgreSQL não implementa]
- (n) : É possível especificar o “tamanho de apresentação”,  
ex: int(5) => 00032 (OBS: faixa de valores não muda)
- (i, d) : para o tipo decimal é possível especificar o número de dígitos totais e decimais,  
ex: decimal(6,2) => 9.999,99 até +9.999,99

# MySQL – Criando Tabelas

- **Outras restrições / atributos opcionais:**

- AUTO\_INCREMENT: indica que será utilizado um valor incrementável automaticamente para o campo (utilizável apenas com tipos numéricos como `INT`, `BIGINT`)
- DEFAULT valor: indica um valor padrão fixo para o campo, que será utilizado caso um valor não seja especificado
- CHECK (predicado): indica que deverá ser realizada uma verificação de integridade de acordo com o predicado.

```
CREATE TABLE Materiais (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) UNIQUE NOT NULL,  
    preco NUMERIC(6,2) CHECK (preco > 0),  
    em_estoque BOOL DEFAULT true  
);
```

# MySQL – Criando Tabelas

- **Exemplo de restrições (formas alternativas)**

```
CREATE TABLE Materiais(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL,  
    preco NUMERIC(6,2) CHECK (preco > 0)  
    preco_vista NUMERIC(6,2),  
    PRIMARY KEY (id),  
-- CONSTRAINT pk PRIMARY KEY (id),  
    UNIQUE (nome),  
-- CONSTRAINT nome_diferente UNIQUE (nome),  
    CHECK (preco_vista <= preco AND preco_vista > 0)  
-- CONSTRAINT avista CHECK  
--     (preco_vista <= preco AND preco_vista > 0)  
);
```

-- => comentário de linha

# MySQL – Criando Tabelas

- **Múltiplas chaves primárias ou estrangeiras:**

```
CREATE TABLE Contatos (  
    cpf NUMERIC(11,0),  
    telefone NUMERIC(11,0),  
    comercial BOOL NOT NULL DEFAULT false,  
    empresa_id INT,  
    PRIMARY KEY (cpf, telefone),  
    FOREIGN KEY cpf REFERENCES Pessoas(cpf),  
    FOREIGN KEY empresa_id  
        REFERENCES Empresas(id),  
);
```

# MySQL – Criando Tabelas

- **Índices**

- São utilizados para acelerar consultas a dados (assim como ocorre em um livro). Entretanto, índices ocupam espaço e devem ser utilizados com parcimônia
  - ❖ Índices são definidos automaticamente em PKs

```
CREATE [UNIQUE] INDEX nome_indice  
    ON nome_tabela (nome_atrib [{, nome_atrib_n}])  
  
DROP INDEX nome_indice ON nome_tabela
```

# MySQL – Modelos de Armazenamento

- ***Storage Engine* ou *Engine***

- São componentes do SGBD que são responsáveis pela realização das operações SQL
- ❑ **PostgreSQL** suporta e utiliza apenas um único *engine* que recebe o seu nome: PostgreSQL
- ❑ **MySQL** suporta e possibilita o uso de diversos engines, entre eles: InnoDB, MyISAM, MEMORY, CSV, ARCHIVE, BLACKHOLE, MERGE, FEDERATED, EXAMPLE

```
CREATE TABLE Disciplinas (  
    [...]  
) engine=MyISAM;
```

```
CREATE TABLE Disciplinas (  
    [...]  
) engine=MyISAM;
```

# MySQL – Modelos de Armazenamento

- ***Storage Engine* ou *Engine***

No MySQL os *engines* mais comuns são:

- ❑ MyISAM: modelo simples e eficiente
  - ✓ **Contras**: não garante restrições de integridade referenciais
  - ✓ **Prós**: simples e bom desempenho;  
bom para muitas leituras e poucas escritas;  
bom quando o controle de restrições está no código
- ❑ InnoDB: modelo mais robusto
  - ✓ **Contras**: um pouco mais lento do que o MyISAM
  - ✓ **Prós**: garante as propriedades ACID;  
bom para escritas concorrentes;
- ❑ Archive: armazena dados em formato compactado, ideal para armazenar dados de históricos que são pouco consultados
- ❑ CSV: armazena dados em formato CSV. Útil para *import/export*
- ❑ Blackhole: descarta todos os registros. Útil para testes e depuração



# MySQL – Modelos de Armazenamento

- Definindo *charset* e *collate*
  - Pode-se definir o `charset` – conjunto de caracteres – e o `collate` – regras de comparação de caracteres – que serão utilizados para as colunas de texto em uma tabela
  - Fundamentais para permitir caracteres especiais de uma língua específica e comparações funcionais (tanto para filtros como para ordenação)

**latin1:** textos em inglês idiomas latinos básicos

**utf8:** suporta a maioria dos idiomas do mundo

**utf8mb4:** pode armazenar emojis e outros caracteres mais complexos

**latin1\_swedish\_ci:** não difere maiúsculas de minúsculas e acentos no charset latin1

**utf8mb4\_unicode\_ci:** não difere maiúsculas de minúsculas e acentos no charset utf8mb4

**utf8mb4\_bin:** difere maiúsculas de minúsculas e acentos no charset utf8mb4

```
CREATE TABLE MinhaTabela (  
  nome VARCHAR(50) CHARACTER SET utf8mb4  
    COLLATE utf8mb4_unicode_ci  
);
```

# MySQL – Modelos de Armazenamento

- Definindo *charset* e *collate*

```
CREATE TABLE Cidade(  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(50) NOT NULL,  
  prefeito VARCHAR(100) NOT NULL,  
) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
```

Define *charset* e *collate* para toda a tabela

Por padrão, o MySQL utiliza o *collate* `utf8mb_0900_ai_ci`, onde  
*ai*=*accent-insensitive* e *ci*=*case-insensitive*  
com ordenação baseada em regras do  
idioma hindi (ISO 15924 código 'hin')

# MySQL – Apagando Tabelas

---

- Para excluir uma tabela, assim como todos os seus dados (se houver) deve-se utilizar o comando:

```
DROP TABLE nome_tabela;
```

# MySQL – Alterando Tabelas

- Existem diversas alterações possíveis que podem ser feitas em uma tabela já existente:

```
ALTER TABLE nome_tabela  
  RENAME TO novo_nome  
  ADD [COLUMN] nome tipo [{Ris}]  
  RENAME [COLUMN] nome_antigo TO nome_novo  
  DROP [COLUMN] nome  
  ALTER [COLUMN] nome [{SET|DROP} DEFAULT  
    | {SET|DROP} NOT NULL | SET DATA TYPE tipo|...]  
  ADD CONSTRAINT ri [PK|FK|CHECK|...]  
  DROP CONSTRAINT nome_ri  
  ...
```

Restrições de integridade

Não esqueça do ponto-e-vírgula ao final do comando!

# MySQL – Alterando Tabelas

- Alguns exemplos de uso:

```
ALTER TABLE alunos  
  RENAME TO graduandos;
```

```
ALTER TABLE graduandos  
  ADD COLUMN dt_nasc  
  DATE NOT NULL;
```

```
ALTER TABLE graduandos  
  RENAME COLUMN dt_nasc  
  TO data_nascimento;
```

```
ALTER TABLE graduandos  
  DROP COLUMN data_nascimento;
```

```
ALTER TABLE graduandos  
  ADD CONSTRAINT nroMat  
  CHECK (matricula > 10000);
```

```
ALTER TABLE graduandos  
  DROP CONSTRAINT nroMat;
```

```
ALTER TABLE disciplinas  
  ALTER COLUMN ch  
  SET DEFAULT (32);
```

<https://dev.mysql.com/doc/refman/8.0/en/alter-table.html>

# MySQL – Sequências em Tabelas

- Os SGBDs implementam formas distintas de uso de números sequenciais em tabelas:
  - **MySQL:** a palavra-chave `auto_increment` cria um campo sequencial para números
    - ✓ simplicidade: mais simples de criar
    - ✓ menos customizável: menos opções de configurações e está ligado a uma tabela
  - **PostgreSQL:** utiliza um objeto próprio chamado `SEQUENCE`
    - ✓ mais customizável: permite mais opções de configurações
    - ✓ múltiplas tabelas: permite compartilhar regra entre tabelas.

```
CREATE TABLE Materiais(  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(50) NOT NULL,  
    PRIMARY KEY (id)  
) AUTO_INCREMENT = 1000;
```

# MySQL – Sequências em Tabelas

- Os SGBDs implementam formas distintas de uso de números sequenciais em tabelas:
  - **PostgreSQL**: utiliza um objeto próprio chamado `SEQUENCE`

```
CREATE SEQUENCE minha_sequencia
START WITH 100
INCREMENT BY 5
MINVALUE 1
MAXVALUE 999
CYCLE;
```

```
CREATE TABLE Materiais(
    id INT PRIMARY KEY
        DEFAULT nextval('minha_sequencia'),
    nome VARCHAR(50) NOT NULL,
);
```

# MySQL – Dump

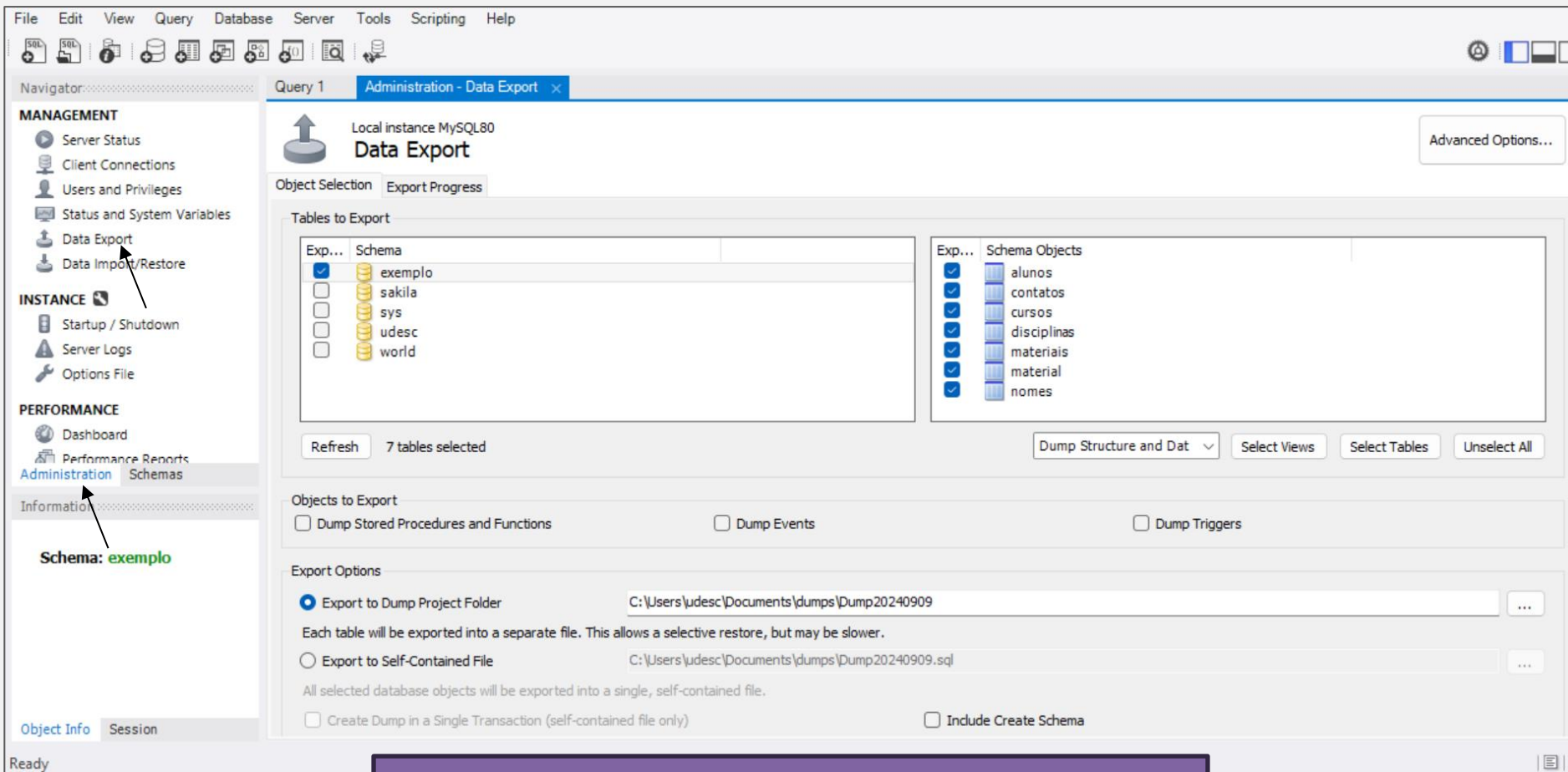
- Para realizar um DUMP (backup ou cópia de segurança) de uma base de dados, podemos utilizar a ferramenta `mysqldump` do MySQL

```
> mysqldump.exe -u root -p exemplo >  
C:\Users\udesc\Downloads\dump-exemplo.sql
```

Considerando que o shell/prompt está no  
diretório bin da instalação do MySQL



# MySQL – Dump



Na GUI, o dump pode ser feito através do painel  
Administration > Data Export

# Atividades

Guarde/salve os comandos SQL. Vamos reutilizar esses dados!

Execute os seguintes procedimentos utilizando apenas comandos do MySQL, sem utilizar a interface gráfica do MySQL Workbench.

1. Crie um BD com o nome **Clinica**
2. Crie 4 novos usuários:
  - a) admin: acesso em localhost e senha \$superAdminCL1N1CA
  - b) suporte: acesso em localhost e senha \$suport3CL1N1CA
  - c) operador: acesso em localhost e senha OperadorCL1N1CA
  - d) consulta: acesso em qualquer host e senha consultaCL1N1CA
3. Conceda a permissão para que o usuário admin tenha todos os privilégios para alterar as tabelas no BD **Clinica**.
4. Faça logout como *root* e acesse o sistema com o usuário admin.

# Atividades

Guarde/salve os comandos SQL. Vamos reutilizar esses dados!

5. Crie as seguintes tabelas neste BD:

**Ambulatorios:** #nroa (int), andar (numeric(3), não nulo)

**Médicos:** #codm (int), nome (varchar(50), não nulo),  
idade (smallint, não nulo), especialidade (char(20)),  
CPF (numeric(11), único), cidade (varchar(30)), nroa (int)

**Pacientes:** #codp (int), nome (varchar(40), não nulo), idade (smallint),  
CPF (numeric(11), único), doença (varchar(40), não nulo)

**Funcionarios:** #codf (int), nome (varchar(40), não nulo),  
idade (smallint), CPF (numeric(11), único), cidade (varchar(30)),  
salario (numeric(10)), cargo (varchar(20))

**Consultas:** #&codm (int), &codp (int), #data (date), #hora (time)

FK – Chaves estrangeiras

OBSERVAÇÃO: faça exatamente o que se pede.  
Algumas “coisas estranhas” serão alteradas posteriormente

# Atividades

Guarde/salve os comandos SQL. Vamos reutilizar esses dados!

6. Conceda as seguintes permissões aos usuários:
  - a) suporte: direitos para selecionar, inserir, atualizar e remover dados em todas as tabelas do BD **Clinica**;
  - b) operador: direitos para selecionar, inserir e atualizar dados nas tabelas **Consultas** e **Pacientes** do BD **Clinica**;
  - c) consulta: direitos para selecionar dados nas tabelas **Consultas** e **Pacientes** do BD **Clinica**.
7. Crie uma restrição de verificação que garanta que não possa ser registrada uma consulta antes das 06:00 nem depois das 21:00.
8. Altere a tabela **Ambulatorios** criando a coluna `capacidade` (SMALLINT) e a tabela **Pacientes** criando a coluna `cidade` (VARCHAR(30)). Ambos os campos são opcionais.

# Atividades

Guarde/salve os comandos SQL. Vamos reutilizar esses dados!

9. Altere a tabela **Funcionarios** removendo a coluna `cargo`
10. Altere a tabela **Medicos**:
  - a. crie uma FK para a coluna `nroa` que faz referência à coluna `nroa` de **Ambulatórios**
  - b. crie uma nova coluna `ativo (bool)` como um campo obrigatório e com valor padrão inicial verdadeiro (`true`)
11. O campo `doenca` foi mal interpretado e está no lugar errado. Corrija removendo este campo na tabela **Pacientes** e adicione-o na tabela **Consultas**
12. Alguns problemas aconteceram e o usuário consulta não deve mais ter privilégios para consultar na tabela de pacientes
13. Realize um dump do BD **Clinica** em um arquivo `clinica-dump.sql`