

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
ENSINO TÉCNICO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS AMS

DANILO SANTOS SOARES

DESENVOLVIMENTO DA ATIVIDADE “MER E DER”

São Paulo

2023

MER E DER

Crie o Der e o Mer da situação abaixo, fazendo o teste de mesa para confirmar a carnalidade definida:

ATIVIDADE PARTE 6:

Um zoológico planeja adquirir um sistema para gerenciar as suas tarefas diárias.

(O sistema para esse zoológico deve ser capaz de cadastrar os seus animais, bem como "os seus funcionários.

O sistema deve, ainda, reconhecer a que classe determinado animal pertence:

mamífero, réptil, ave. Além disso, cada classe deve conter uma descrição técnica sobre as suas características específicas.

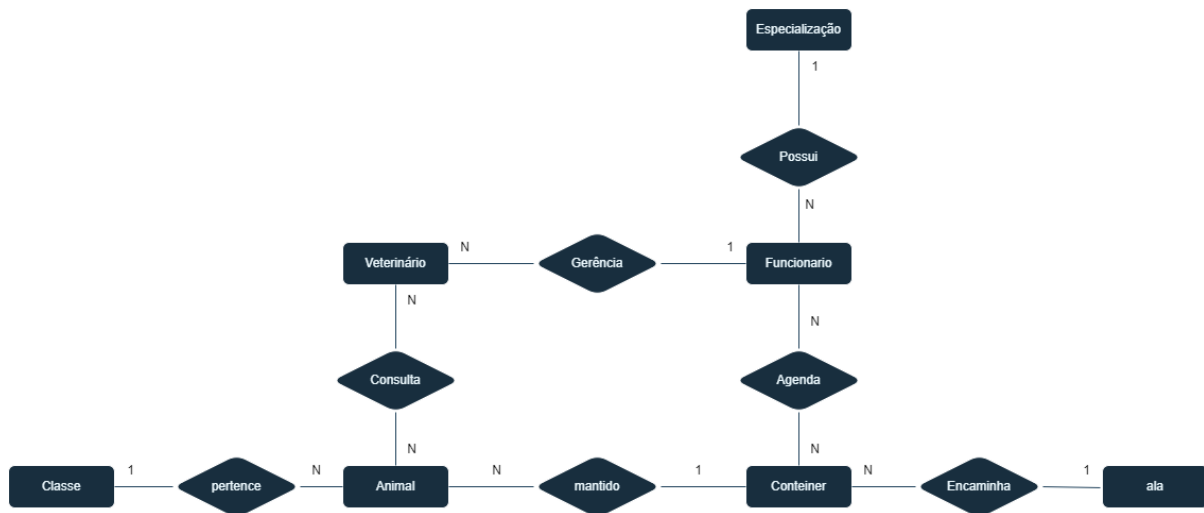
Os animais do zoológico devem ser identificados por um código de identificação, devendo o sistema registrar o seu nome, espécie, cor e altura. Cada animal é mantido em algum tipo de container, e o sistema deve saber o seu tipo, por exemplo: um poço, uma jaula, um viveiro, um tanque, etc. Cada container fica localizado em uma ala do Zoológico, e o sistema deve indicar qual é a ala onde ele se encontra para facilitar o agendamento de atividades.

Os funcionários que trabalham nesse zoológico podem ser veterinários, cuidadores de animais, zeladores ou trabalhareem em setores administrativos. Todos os funcionários devem conter nome, data de nascimento, CPF, RG, endereço completo e O cargo que desempenham. No caso dos veterinários, deve conter ainda o CRMV. (Carteira do Conselho Regional de Medicina Veterinária).

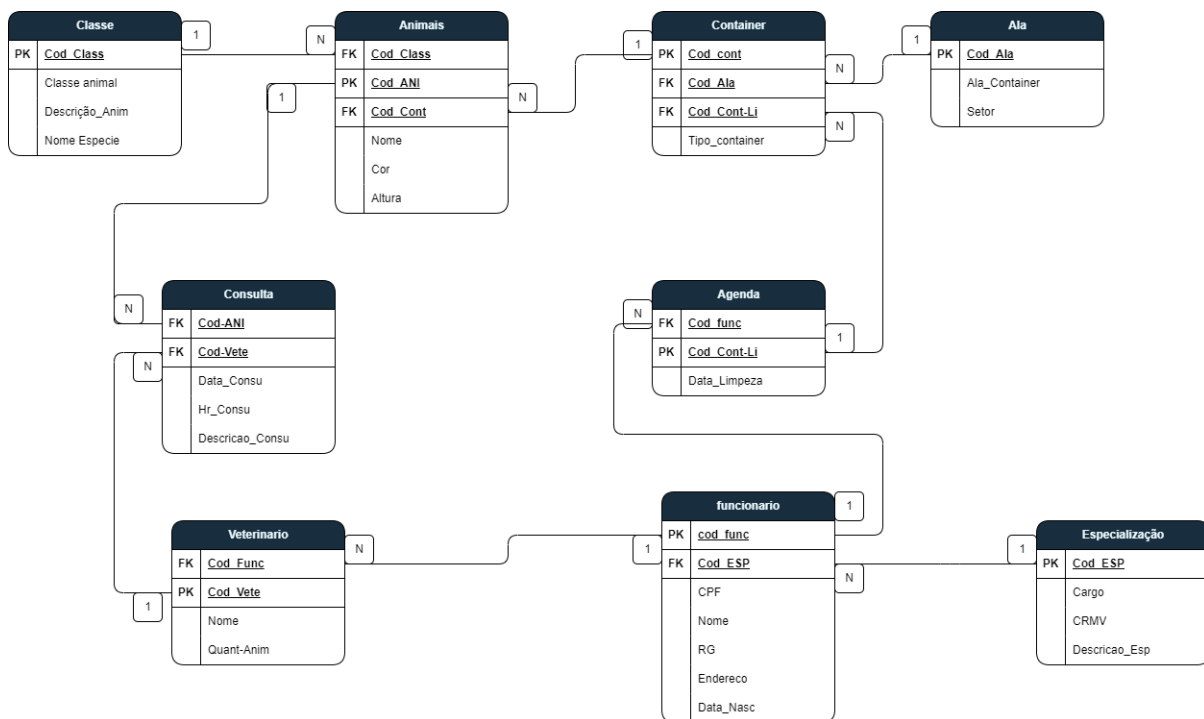
|O sistema deve possuir um módulo para agendar dia e hora para as consultas de cada animal com os veterinários, tendo em vista que cada animal deve ser tratado sempre pelo mesmo veterinário, para que ele possa fazer acompanhamentos em longo prazo com o seu paciente, ressaltando que um veterinário pode atender, no máximo, 15 animais, sendo este o número ideal para que o veterinário consiga manter a qualidade do seu trabalho.

O sistema deve também ser capaz de agendar dia e horário específico para a limpeza dos containers dos animais.

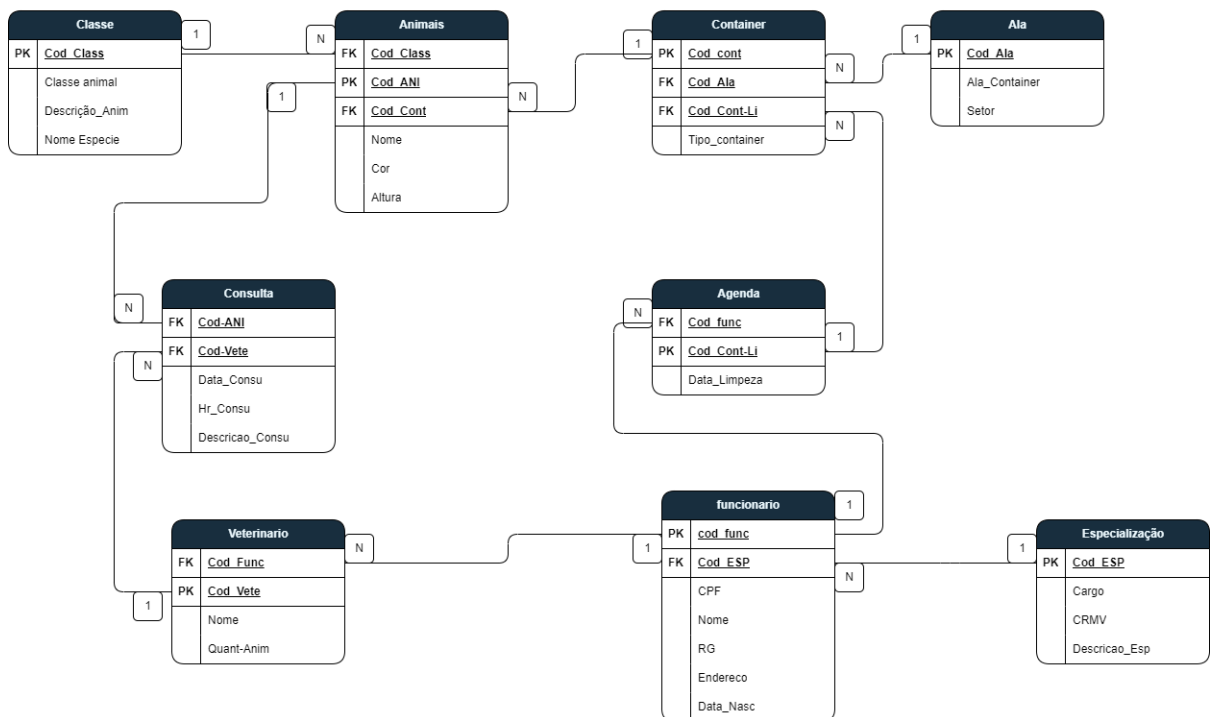
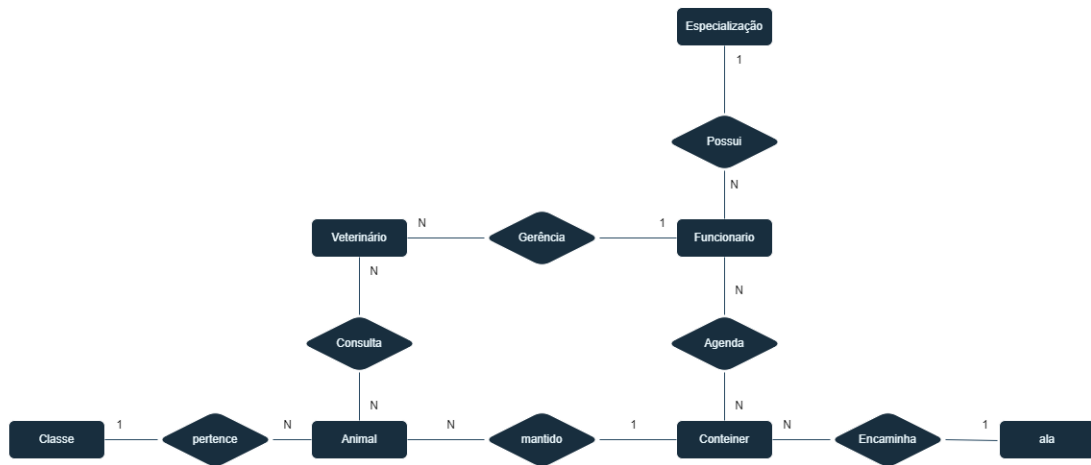
DER:

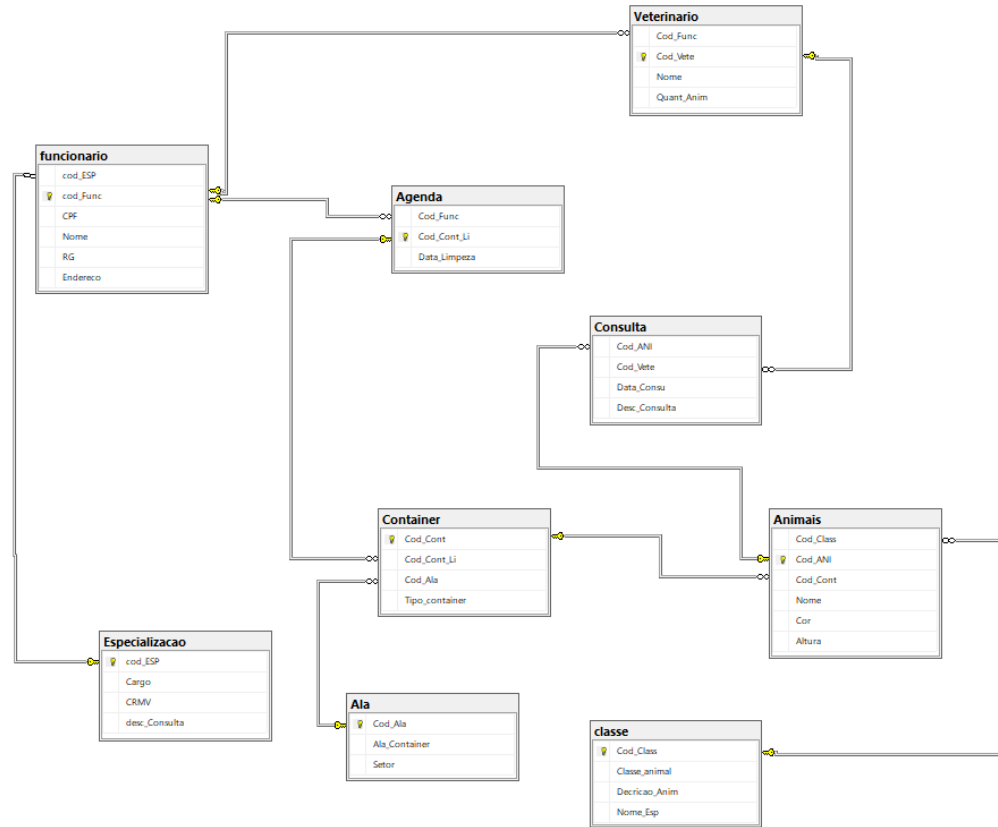


Mer:

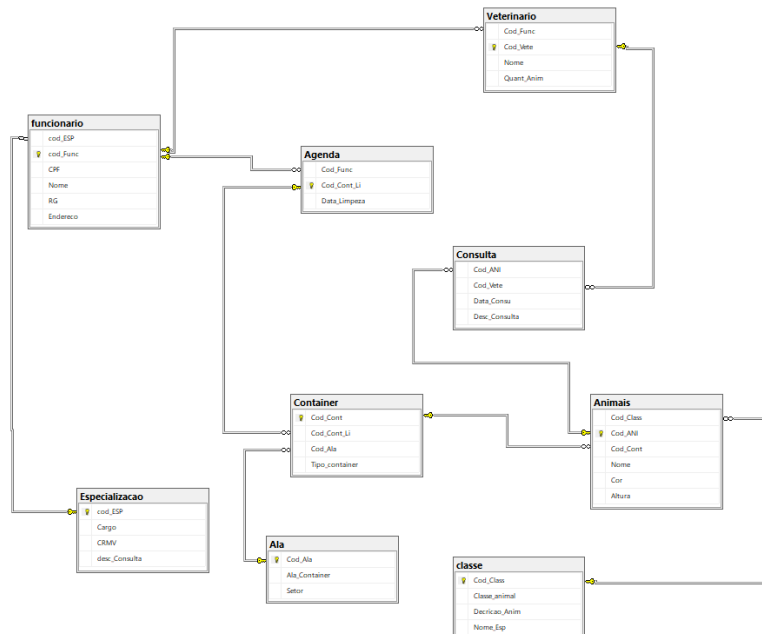


Final:





- Conectar
- Desconectar
- Bancos de Dados
- Bancos de Dados do Sistema
- Instantâneos do Banco de Dados
- Diagramas de Banco de Dados
- Tabelas
- Exibições
- Recursos Externos
- Sinônimos
- Programação
- Service Broker
- Armazenamento
- Segurança
- Objetos de Servidor
- Replicação
- PolyBase
- Gerenciamento
- XEEvent Profiler



Dicionário de Dado

Tabela	Agenda			
Descrição	Armazenar as informações de Agenda			
Observações	Tabela associada com a tabela Container tendo como função ser PK e ser Fk na tabela funcionário			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_Func	Número do Funcionário.	INT		(Fk)
Cod_Cont_li	Número do container para limpeza sendo unico para cada	Int		(PK) Identity
Data_Limp	Data da Limpeza e o horário	Datetime		Not null

Tabela	Ala			
Descrição	Armazenar as informações de ala			
Observações	Tabela associada com a tabela Container tendo como função ser PK			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_Ala	Número da Ala sendo unico para cada.	INT		(Pk)
Ala_Container	Ala em que o container está localizado	Vachar	100	Not null
Setor	Setor em que Ala se encontra	Vachar	6	Not null

Tabela	Container			
Descrição	Armazenar as informações de Container			
Observações	Tabela associada com a tabela Agenda e Ala tendo como função ser FK e PK da tabela Animais			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_Cont	Número o Container sendo unico para cada.	INT		(Pk) Identity
Cod_Cont li	Número do Container a ser registrado	Int		(FK)
Cod_Ala	Número do Ala a ser registrado	INT		(FK)
Tipo container	Tipo do container para armazenar animais	Vachar	50	Not null

Tabela	Classe			
Descrição	Armazenar as informações de Classe			
Observações	Tabela associada com a tabela Animais tendo como função ser PK			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_Class	Número da Classe sendo unico para cada.	INT		(PK)Identity
Class_Animal	Tipo de Classe do Animal	Vachar	50	Not null
Desc_Animal	Descrição do Animal	Varchar	60	Not null
Nome_EPS	Nome da Especialização a ser registrado	Vachar	60	Not null

Tabela	Animais			
Descrição	Armazenar as informações de animais			
Observações	Tabela associada com a tabela classe e Container tendo como função ser FK e ser PK da tabela Consulta			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_Class	Número da Class.	INT		(Fk)
Cod_Cont	Número do Container sendo unico para cada.	INT		(FK)
Cod_ANI	Número da Animal sendo unico para cada.	INT		(Pk) Identity
Nome	Nome do Animal a ser registrado	Varchar	100	Not null
Altura	Altura do animal	Float		Not null
Cor	Cor do Animal	Vachar	50	Not null

Tabela	Veterinário			
Descrição	Armazenar as informações de Veterinário			
Observações	Tabela associada com a tabela Consulta tendo como função ser PK e ser FK da tabela funcionário			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_Func	Número do Funcionário.	INT		(Fk)
Cod_Vete	Número do Veterinário sendo Único para cada	INT		(PK) Identity
Nome	Nome do Veterinário a ser registrado	Varchar	100	not null
Quant_Animais	Quantidade armazenada de ANI para veterinário	INT		Not null

Tabela	Consulta			
Descrição	Armazenar as informações da Consulta			
Observações	Tabela associada com a tabela Animais e Veterinário tendo como função ser FK			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod ANI	Número do Código dos Animais	INT		(Fk)
Cod Vete	Nome do aluno a ser registrado	INT		(FK)
Data Cons	Data e Hora da Consulta	DATETIME		Not null
Desc Consulta	Descrição da Consulta	Vachar	50	Not null

Tabela	Especialização			
Descrição	Armazenar as informações de Especialização			
Observações	Tabela associada com a tabela funcionario tendo como função ser PK			
		Campos		
	Descrição	Tipo de Dado	Tamanho	Restrições de dominio (PK,FK,Not Null,Check,Default)
Cod_ESP	Código de Especialização sendo unico para cada.	INT		(pk) Identity
CRMV	Conselho Federativa de medicina veterinária	Varchar	10	Not null
Desc_Cosulta	Descrição da Consulta	Varchar	80	Not null
Cargo	cargo do funcionario	varchar	50	Not null

Tabela	Funcionário			
Descrição	Armazenar as informações do Funcionário			
Observações	Tabela associada com a tabela Especialização tendo como função ser FK e ser Pk da tabela Agenda			
Campos				
	Descrição	Tipo de Dado	Tamanho	Restrições de domínio (PK,FK,Not Null,Check,Default)
Cod_ESP	Código de Especialização .	INT		(Fk)
Nome	Nome do Funcionário a ser registrado	Vachar	100	Not null
CPF	Documento de cadastro de pessoa física	Varchar	13	Not null
RG	Carteira de Identidade do funcionário	Varchar	7	Not null
Endereço	Local de residencia do funcionario	Varchar	20	Not null
Cod_Func	Código do Funcionário sendo unico para cada	INT		(PK) Identity

Código fonte do BD:

PARTE 6 ZOO:

```
create database zoo;
```

```
use zoo;
```

```
create table Especializacao(  
cod_ESP int identity,  
Cargo varchar(50)not null,  
CRMV varchar (10) NOT NULL,  
desc_Consulta varchar (80) not null,
```

```
primary key(cod_ESP),  
);
```

```
create table funcionario(  
cod_ESP INT,  
cod_Func Int identity,  
CPF VARCHAR (13) NOT NULL,  
Nome varchar (50) not null,  
RG varchar (11) not null,  
Endereco varchar(60)not null,
```

```
primary key(cod_Func),  
FOREIGN KEY (Cod_ESP) REFERENCES Especializacao(Cod_ESP),  
  
);
```

```
create table Agenda(  
    Cod_Func int ,  
    Cod_Cont_Li int identity,  
    Data_Limpeza datetime,  
  
    primary key(Cod_Cont_Li),  
  
    FOREIGN KEY (Cod_Func) REFERENCES funcionario(Cod_Func),  
);
```

```
create table Ala(  
    Cod_Ala int identity,  
    Ala_Container Varchar (20) not null,  
    Setor varchar(10) not null,  
  
    primary key(Cod_Ala),  
);
```

```
create table Container(  
    Cod_Cont int identity,  
    Cod_Cont_Li int ,  
    Cod_Ala int,  
    Tipo_container varchar(20),  
  
    primary key(Cod_Cont),  
  
    FOREIGN KEY (Cod_Cont_Li) REFERENCES Agenda(Cod_Cont_Li),  
    FOREIGN KEY (Cod_Ala) REFERENCES Ala(Cod_Ala),  
);
```

```
create table classe(  
Cod_Class Int identity,  
Classe_animal varchar (50) not null,  
Decricao_Anim varchar (60)not null,  
Nome_Esp varchar (60)not null,  
  
primary key (Cod_Class),  
  
);
```

```
create table Animais(  
Cod_Class Int,  
Cod_ANI Int identity,  
Cod_Cont Int,  
Nome varchar(50) not null,  
Cor Varchar (60) Not null,  
Altura float Not null,  
  
primary key(Cod_ANI),  
  
FOREIGN KEY (Cod_Class) REFERENCES Classe(Cod_Class),  
FOREIGN KEY (Cod_Cont) REFERENCES Container(Cod_Cont),  
);
```

```
create table Veterinario(  
Cod_Func int,  
Cod_Vete int identity,  
Nome varchar (60) not null,  
Quant_Anim int,
```

```
primary key(Cod_vete),
```

```
FOREIGN KEY (Cod_Func) REFERENCES funcionario(Cod_Func),  
);
```

```
create table Consulta(  
Cod_ANI int,  
Cod_Vete int,  
Data_Consu DATETIME,  
Desc_Consulta varchar(100) not null,
```

```
FOREIGN KEY (Cod_Vete) REFERENCES Veterinario(Cod_Vete),  
FOREIGN KEY (Cod_ANI) REFERENCES Animais(Cod_ANI),  
);
```

```
INSERT INTO Especializacao (Cargo, CRMV, desc_Consulta)  
VALUES  
('Veterinário de Mamíferos', 'CRMV123', 'Consulta regular de mamíferos'),  
('Veterinário de Aves', 'CRMV456', 'Consulta regular de aves'),  
('Veterinário de Répteis', 'CRMV789', 'Consulta regular de répteis');
```

```
INSERT INTO Funcionario (cod_ESP, CPF, Nome, RG, Endereco)  
VALUES  
(1, '12345678901', 'João Silva', '1234567890', 'Rua A, 123'),  
(2, '23456789012', 'Maria Souza', '2345678901', 'Avenida B, 456'),
```

```
(3, '34567890123', 'Pedro Santos', '3456789012', 'Praça C, 789');
```

```
INSERT INTO Agenda (Cod_Func, Data_Limpeza)
```

```
VALUES
```

```
(1, '2023-11-27 09:00:00'),
```

```
(2, '2023-11-28 10:30:00'),
```

```
(3, '2023-11-29 08:45:00');
```

```
INSERT INTO Ala (Ala_Container, Setor)
```

```
VALUES
```

```
('Jaula', 'Setor A'),
```

```
('Viveiro', 'Setor B'),
```

```
('Tanque', 'Setor C');
```

```
INSERT INTO Container (Cod_Cont_Li, Cod_Ala, Tipo_container)
```

```
VALUES
```

```
(1, 1, 'Pequeno'),
```

```
(2, 2, 'Grande'),
```

```
(3, 3, 'Médio');
```

```
INSERT INTO Classe (Classe_animal, Decricao_Anim, Nome_Esp)
```

```
VALUES
```

```
('Mamífero', 'Mamíferos selvagens', 'Espécie X'),
```

```
('Ave', 'Aves exóticas', 'Espécie Y'),
```

```
('Réptil', 'Répteis aquáticos', 'Espécie Z');
```

```
INSERT INTO Animais (Cod_Class, Cod_Cont, Nome, Cor, Altura)
```

```
VALUES
```

```
(1, 1, 'Leão', 'Marrom', 1.2),
```

```
(2, 2, 'Arara Azul', 'Azul', 0.5),
```

```
(3, 3, 'Tartaruga', 'Verde', 0.3);
```

```
INSERT INTO Veterinario (Cod_Func, Nome, Quant_Anim)
```

```
VALUES
```

```
(1, 'Dr. Carlos', 5),
```

```
(2, 'Dra. Ana', 8),
```

```
(3, 'Dr. Rafael', 4);
```

```
INSERT INTO Consulta (Cod_ANI, Cod_Vete, Data_Consu, Desc_Consulta)
```

```
VALUES
```

```
(1, 1, '2023-12-01 10:00:00', 'Consulta de rotina para leão'),
```

```
(2, 2, '2023-12-02 11:30:00', 'Exame de saúde para arara azul'),
```

```
(3, 3, '2023-12-03 09:45:00', 'Tratamento para tartaruga');
```

```
select * from Agenda;
```

```
select * from Container;
```

```
select * from Especializacao;
```

```
select * from funcionario;
```

```
select * from Animais;
```

```
select * from Veterinario;
```

```
select * from Consulta;
```

```
select * from Ala;
```

```
select * from classe;
```

drop table Consulta;

drop table Veterinario;

drop table Animais;

drop table classe;

drop table Container;

drop table Ala;

drop table Agenda;

drop table funcionario;

drop table Especializacao;

drop database z