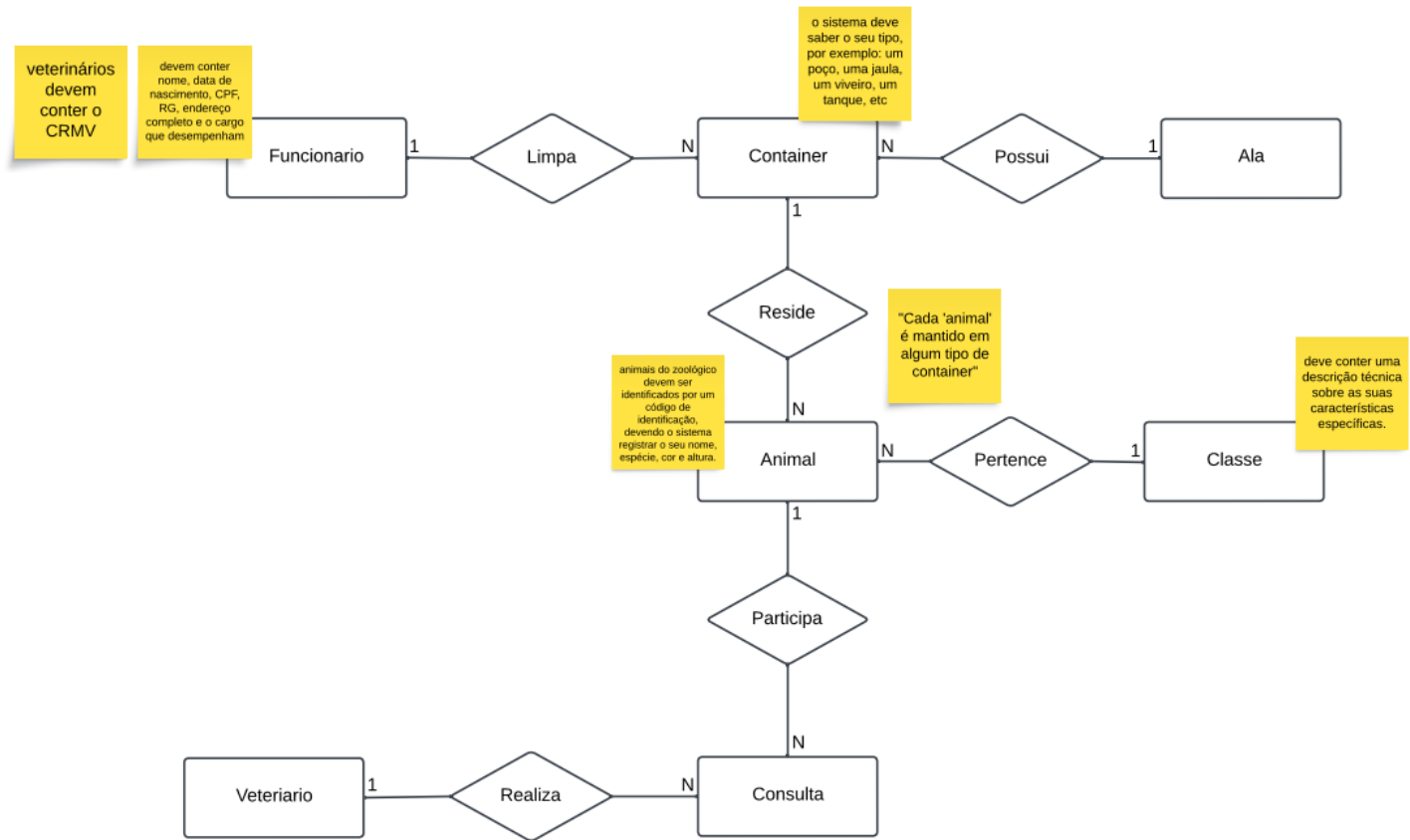


Documentação Zoológico

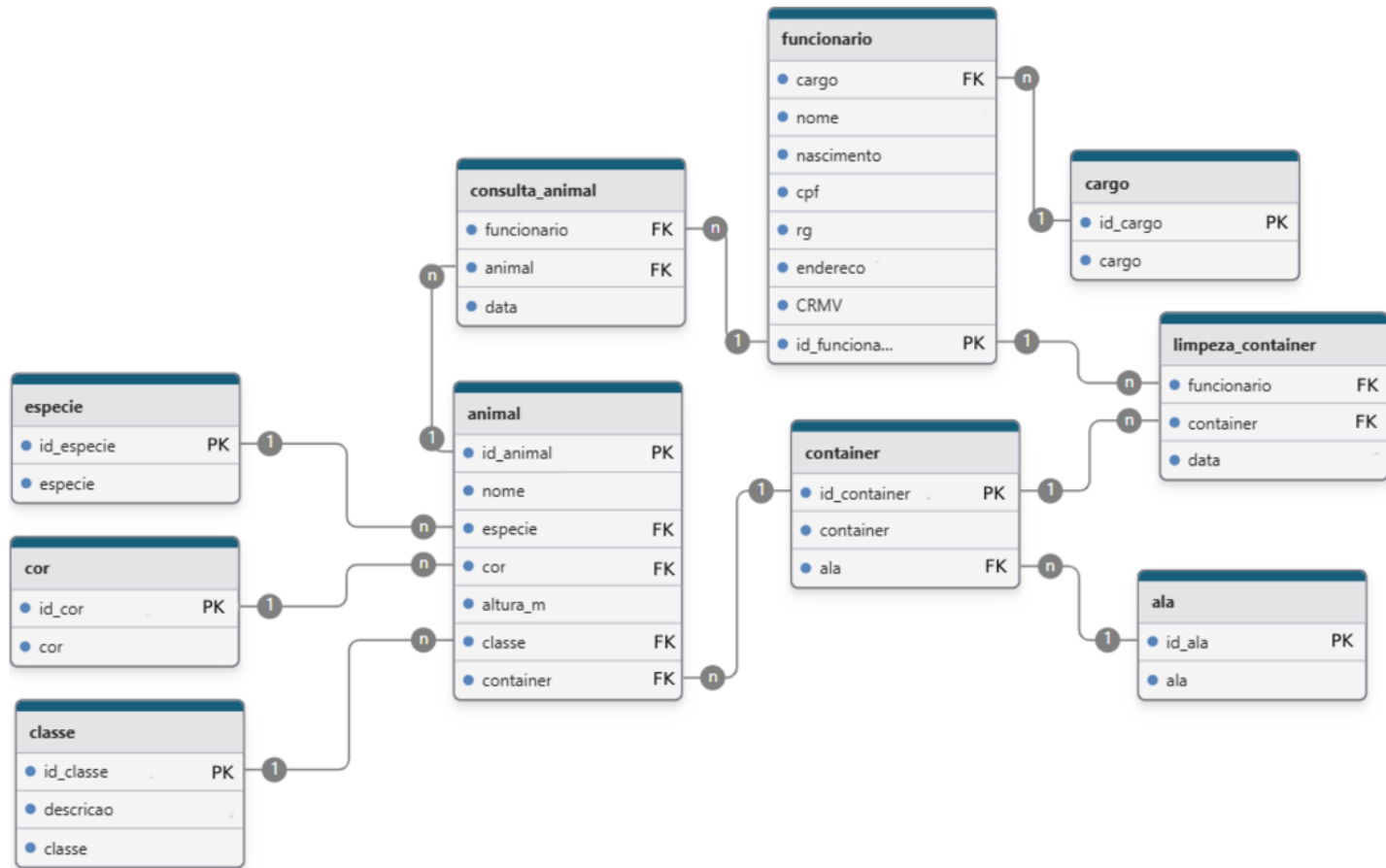
Sumário

DER.....	2
MER	3
Dicionário de dados	4
‘especie’	4
‘cor’	4
‘classe’	5
‘animal’	5
‘consulta_animal’	6
‘funcionario’	7
‘cargo’	8
‘container’	8
‘limpeza_container’	9
‘ala’	9
Código SQL.....	10
MER SSMS	13

DER



MER



Dicionário de dados

Tabelas:

‘especie’

Tabela:	especie			
Descrição:	Armazena dados da espécies de animais possíveis			
Observação:	Possui PK que se liga com a tabela 'animal'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_especie	Código único de especie	Int		PK
especie	Especie do animal	Varchar	255	Not Null

‘cor’

Tabela:	cor			
Descrição:	Armazena cores possíveis dos animais			
Observação:	Possui PK que se liga com a tabela 'animal'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_cor	Código único de cor	Int		PK
cor	Cor do animal	Varchar	255	Not Null

‘classe’

Tabela:	classe			
Descrição:	Armazena a classe a qual o animal pertence			
Observação:	Possui PK que se liga com a tabela 'animal'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_classe	Código único de classe	Int		PK
descricao	Descrição da classe	Varchar	255	Not Null
classe	Nome da classe	Varchar	255	Not Null

‘animal’

Tabela:	animal			
Descrição:	Armazena as informações sobre cada animal			
Observação:	Possui PK que se liga com a tabela 'consulta_animal' e FK's provenientes das tabelas 'especie', 'cor', 'classe' e 'container' respectivamente			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_animal	Código único do animal	Int		PK
nome	Nome do animal	Varchar	50	Not Null
especie	Código da espécie	Int		FK
cor	Código da cor	Int		FK
altura_m	Altura do animal em metros	Int		Not Null
classe	Código da classe	Int		FK
container	Código do container	Int		FK

‘consulta_animal’

Tabela:	consulta_animal			
Descrição:	Armazena dados das consultas dos animais			
Observação:	Possui duas FK's proveniente das tabelas 'funcionario' e 'animal' respectivamente			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
funcionario	Código do funcionário	Int		FK
animal	Código do animal	Int		FK
data	Data da consulta	Date		Not Null

‘funcionario’

Tabela:	funcionario			
Descrição:	Armazena informações dos funcionários			
Observação:	Possui PK que se liga a tabelas 'consulta_animal' e 'limpeza_container', além de uma FK proveniente da tabela 'cargo'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
cargo	Código do cargo	Int		FK
nome	Nome do funcionário	Varchar	50	Not Null
nascimento	Data de nascimento do funcionário	Date		Not Null
cpf	Cadastro de Pessoas Físicas do funcionário	Varchar	11	Not Null
rg	RG do funcionário	Varchar	9	Not Null
endereco	Endereço do funcionário	Varchar	255	Not Null
CRMV	Número de identificação do Conselho Regional de Medicina Veterinária	Varchar	50	
id_funcionario	Código único de identificação	Int		PK

‘cargo’

Tabela:	cargo			
Descrição:	Armazena os cargos possíveis dos funcionários do zoológico			
Observação:	Possui PK que se liga com a tabela 'funcionario'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_cargo	Código único de identificação	Int		PK
cargo	Nome do cargo	Varchar	20	Not Null

‘container’

Tabela:	container			
Descrição:	Armazena informações dos containers do Zoológico			
Observação:	Possui PK que se liga com as tabelas 'limpeza_container' e 'animal', além de FK proveniente da tabela 'ala'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_container	Código único de identificação	Int		PK
container	Descrição do tipo de container	Varchar	255	Not Null
ala	Código de identificação da ala	Int		FK

‘limpeza_container’

Tabela:	limpeza_container			
Descrição:	Armazena informações sobre a limpeza dos containers			
Observação:	Possui FK's provenientes das tabelas 'funcionario' e 'container' respectivamente			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
funcionario	Código do funcionário	Int		FK
container	Código do container	Int		FK
data	Data da limpeza	Date		Not Null

‘ala’

Tabela:	ala			
Descrição:	Armazena alas do zoológico			
Observação:	Possui PK que se liga com a tabela 'container'			
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrição de domínio
id_ala	Código único da ala	Int		PK
ala	Nome da ala	Varchar	255	Not Null

Código SQL

```
CREATE DATABASE zoologico;  
GO
```

```
USE zoologico;  
GO
```

```
CREATE TABLE [especie] (  
    [id_especie] INTEGER NOT NULL PRIMARY KEY,  
    [especie] VARCHAR NOT NULL  
);  
GO
```

```
CREATE TABLE [classe] (  
    [id_classe] INTEGER NOT NULL PRIMARY KEY,  
    [descricao] VARCHAR NOT NULL,  
    [classe] VARCHAR NOT NULL  
);  
GO
```

```
CREATE TABLE [cor] (  
    [id_cor] INTEGER NOT NULL PRIMARY KEY,  
    [cor] VARCHAR NOT NULL  
);  
GO
```

```
CREATE TABLE [ala] (  
    [id_ala] INTEGER NOT NULL PRIMARY KEY,  
    [ala] VARCHAR NOT NULL  
);  
GO
```

```
CREATE TABLE [carga] (  
    [id_carga] INTEGER NOT NULL PRIMARY KEY,  
    [carga] VARCHAR NOT NULL  
);  
GO
```

```
CREATE TABLE [container] (  
    [id_container] INTEGER NOT NULL PRIMARY KEY,  
    [container] VARCHAR NOT NULL,
```

```
    [ala] INTEGER NOT NULL,  
    FOREIGN KEY([ala]) REFERENCES [ala]([id_ala])  
);  
GO
```

```
CREATE TABLE [animal] (  
    [id_animal] INTEGER NOT NULL PRIMARY KEY,  
    [nome] VARCHAR NOT NULL,  
    [especie] INTEGER NOT NULL,  
    [cor] INTEGER NOT NULL,  
    [altura_m] INTEGER NOT NULL,  
    [classe] INTEGER NOT NULL,  
    [container] INTEGER NOT NULL,  
    FOREIGN KEY([especie]) REFERENCES [especie]([id_especie]),  
    FOREIGN KEY([classe]) REFERENCES [classe]([id_classe]),  
    FOREIGN KEY([cor]) REFERENCES [cor]([id_cor]),  
    FOREIGN KEY([container]) REFERENCES [container]([id_container])  
);  
GO
```

```
CREATE TABLE [funcionario] (  
    [id_funcionario] INTEGER NOT NULL PRIMARY KEY,  
    [cargo] INTEGER NOT NULL,  
    [nome] VARCHAR NOT NULL,  
    [nascimento] DATE NOT NULL,  
    [cpf] VARCHAR NOT NULL,  
    [rg] VARCHAR NOT NULL,  
    [endereco] VARCHAR NOT NULL,  
    [CRMV] VARCHAR,  
    FOREIGN KEY([cargo]) REFERENCES [cargo]([id_cargo])  
);  
GO
```

```
CREATE TABLE [limpeza_container] (  
    [funcionario] INTEGER NOT NULL,  
    [container] INTEGER NOT NULL,  
    [data] DATE NOT NULL,  
    FOREIGN KEY([funcionario]) REFERENCES [funcionario]([id_funcionario]),  
    FOREIGN KEY([container]) REFERENCES [container]([id_container])  
);  
GO
```

```
CREATE TABLE [consulta_animal] (  
    [id_animal] INTEGER NOT NULL PRIMARY KEY,  
    [data] DATE NOT NULL,  
    [hora] TIME NOT NULL,  
    [funcionario] INTEGER NOT NULL,  
    FOREIGN KEY([id_animal]) REFERENCES [animal]([id_animal]),  
    FOREIGN KEY([funcionario]) REFERENCES [funcionario]([id_funcionario])  
);  
GO
```

```
[funcionario] INTEGER NOT NULL,  
[animal] INTEGER NOT NULL,  
[data] DATE,  
FOREIGN KEY([funcionario]) REFERENCES [funcionario]([id_funcionario]),  
FOREIGN KEY([animal]) REFERENCES [animal]([id_animal])  
);  
GO
```

MER SSMS

