

Minimundo

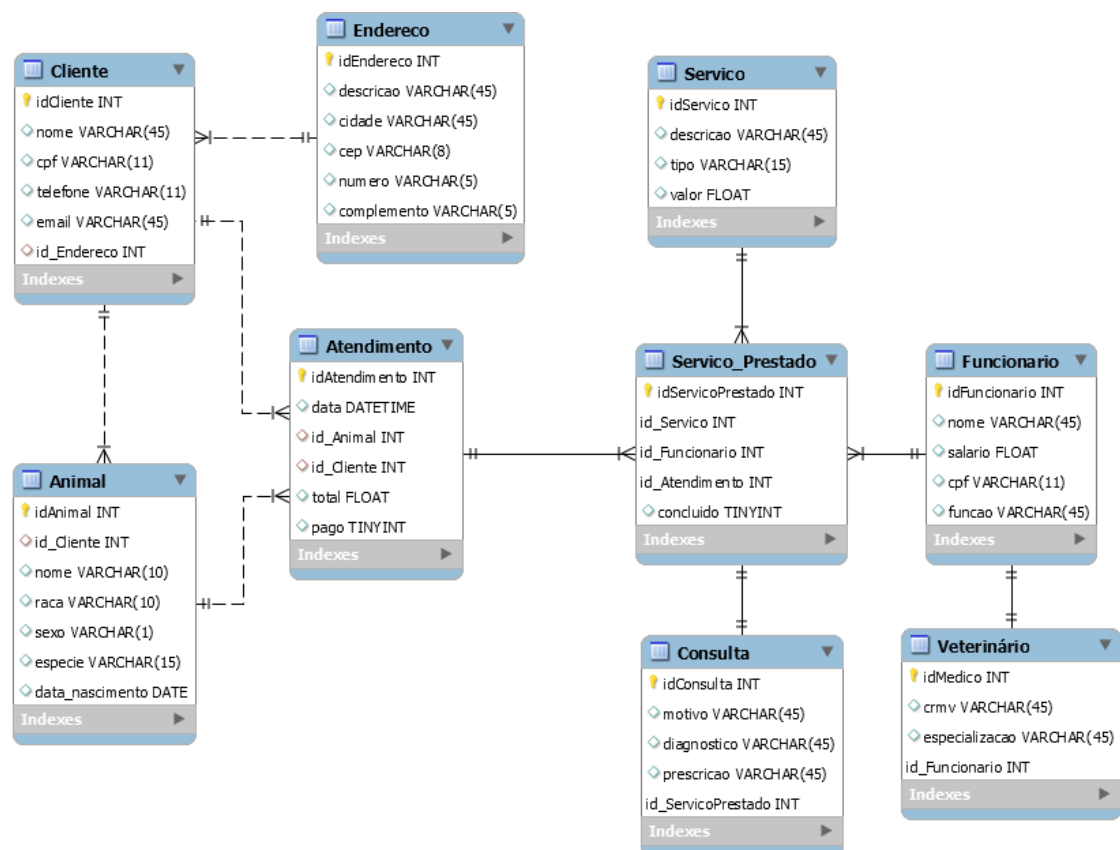
Em uma clínica veterinária, o secretário cadastra clientes com nome, cpf, email, telefone e endereço completo. Funcionários são cadastrados pelo gerente, com nome, cpf, salário e função. Se for médico veterinário, terá também crmv e especialização. Todos os dados cadastrais podem ser alterados por sócios.

Os clientes podem ter um ou mais animais, que por sua vez são cadastrados com nome, espécie, raça, sexo e data de nascimento.

Quando o cliente leva seu animal à clínica, um atendimento é iniciado. Cada atendimento pode incluir diferentes tipos de serviços: exame, vacinação ou consulta. O serviço é prestado por um funcionário, e se for uma consulta será registrado o motivo da mesma. Ao término da consulta, serão registrados ainda a prescrição e o diagnóstico.

Após os serviços serem prestados, o pagamento é recolhido, finalizando assim o atendimento.

Diagrama EER



Comandos DDL:

```
CREATE DATABASE clinica;
```

```
USE clinica;
```

```
CREATE TABLE Endereco (
```

```
    idEndereco INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
```

```
    descricao VARCHAR(45),
```

```
    cidade VARCHAR(45) NOT NULL,
```

```
    cep VARCHAR(9) NOT NULL,
```

```
    numero VARCHAR(5),
```

```
    complemento VARCHAR(10)
```

```
);
```

```
CREATE TABLE Cliente (
```

```
    idCliente INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
```

```
    nome VARCHAR(45),
```

```
    cpf VARCHAR(14),
```

```
    telefone VARCHAR(11),
```

```
    id_Endereco INT,
```

```
    email VARCHAR(45),
```

```
    FOREIGN KEY (id_Endereco) REFERENCES Endereco (idEndereco)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE
```

```
);
```

```
CREATE TABLE Animal (
```

```
    idAnimal INT PRIMARY KEY AUTO_INCREMENT,
```

```
    id_Cliente INT,
```

```
    nome VARCHAR(10) NOT NULL,
```

```
    raca VARCHAR(10) NOT NULL,  
    sexo VARCHAR(1) NOT NULL,  
    especie VARCHAR(15) NOT NULL,  
    data_nascimento DATE NOT NULL,  
    FOREIGN KEY (id_Cliente) REFERENCES Cliente (idCliente)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE Atendimento (  
    idAtendimento INT PRIMARY KEY AUTO_INCREMENT,  
    data DATE NOT NULL,  
    id_Animal INT,  
    id_Cliente INT,  
    total FLOAT NOT NULL,  
    pago TINYINT NULL DEFAULT 0,  
    FOREIGN KEY (id_Cliente) REFERENCES Cliente (idCliente)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE,  
    FOREIGN KEY (id_Animal) REFERENCES Animal (idAnimal)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE Servico (  
    idServico INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    descricao VARCHAR(45) NOT NULL,  
    tipo VARCHAR(15) NOT NULL,  
    valor FLOAT NOT NULL  
);
```

```
CREATE TABLE Funcionario (  
    idFuncionario INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(45) NOT NULL,  
    salario FLOAT NOT NULL,  
    cpf VARCHAR(11) NOT NULL,  
    funcao VARCHAR(45) NOT NULL  
);
```

```
CREATE TABLE Servico_Prestado (  
    idServicoPrestado INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    id_Servico INT,  
    concluido TINYINT DEFAULT 0,  
    id_Funcionario INT,  
    id_Atendimento INT NOT NULL,  
    FOREIGN KEY (id_Servico) REFERENCES Servico (idServico)  
ON DELETE SET NULL  
ON UPDATE CASCADE,  
    FOREIGN KEY (id_Atendimento)  
REFERENCES Atendimento (idAtendimento)  
ON DELETE RESTRICT  
ON UPDATE CASCADE,  
    FOREIGN KEY (id_Funcionario)  
REFERENCES Funcionario (idFuncionario)  
ON DELETE SET NULL  
ON UPDATE CASCADE  
);
```

```
CREATE TABLE Veterinario (  
    idVeterinario INT PRIMARY KEY AUTO_INCREMENT,  
    crmv VARCHAR(9) NOT NULL,
```

```

        especializacao VARCHAR(45),
        id_Funcionario INT NOT NULL,
        FOREIGN KEY (id_Funcionario) REFERENCES Funcionario (idFuncionario)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
    );

```

```

CREATE TABLE Consulta (
    idConsulta INT PRIMARY KEY AUTO_INCREMENT,
    motivo VARCHAR(45),
    diagnostico VARCHAR(45),
    prescricao VARCHAR(45),
    id_ServicoPrestado INT NOT NULL,
    FOREIGN KEY (id_ServicoPrestado)
    REFERENCES Servico_Prestado (idServicoPrestado)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

```

CREATE INDEX idxAnimal ON Animal(data_nascimento);

```

Comandos DML

```

insert into Endereco(descricao, cidade, cep, numero, complemento) values ('Rua 1,
Centro','Campos dos Goytacazes' , '11111111', '10', null);

```

```

insert into Endereco(descricao, cidade, cep, numero, complemento) values ('Avenida 1, Parque
Imperial','Campos dos Goytacazes' , '55555555', '503', '704');

```

```

insert into Endereco(descricao, cidade, cep, numero, complemento) values ('Travessa 1,
Turfe','Campos dos Goytacazes' , '77777777', '347', '12');

```

```

insert into Endereco(descricao, cidade, cep, numero, complemento) values ('Avenida Do
Contorno, Vila Nova','Cabo Frio' , '55555555', '30', '204');

```

insert into Cliente(nome, cpf, telefone, id_Endereco, email) values ('Ricardo dos Santos','11100033366', '22999887766', 1, 'ricsantos@hotmail.com');

insert into Cliente(nome, cpf, telefone, id_Endereco, email) values ('Icaro da Silva','22244466688', '22988003344', 2, 'icasilva@gmail.com');

insert into Cliente(nome, cpf, telefone, id_Endereco, email) values ('Joao Neves','11133355577', '22991772211', 3, 'joao@gmail.com');

insert into Cliente(nome, cpf, telefone, id_Endereco, email) values ('Marina Andrade','89633355577', '22991772244', 4, 'marinaadr@gmail.com');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (1, 'Thor', 'Poodle','M', 'Cachorro', '2019-01-16');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (1, 'Toto', 'Pinscher','M', 'Cachorro', '2011-02-16');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (1, 'Luna', 'Yorkshire','F', 'Cachorro', '2009-07-28');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (2, 'Gatito', 'siames','M', 'Gato', '2014-07-28');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (2, 'Lina', 'Persa','F', 'Gato', '2012-07-20');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (3, 'Fedelho', 'Ingles','M', 'Gato', '2013-09-28');

insert into Animal(id_Cliente, nome, raca, sexo, especie, data_nascimento) values (4, 'Sky', 'Husky','F', 'Cachorro', '2015-04-10');

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2018-07-29', 1, 1, 0);

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2019-05-15', 5, 2, 0);

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2019-07-31', 6, 3, 0);

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2019-01-31', 7, 4, 0);

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2018-03-19', 2, 1, 0);

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2018-01-19', 3, 1, 0);

insert into Atendimento(data, id_Animal, id_Cliente, total) values ('2019-05-22', 4, 2, 0);

insert into Servico (descricao, tipo, valor) values ('V10', 'vacina', 120.00);

insert into Servico (descricao, tipo, valor) values ('Antirrabica', 'vacina', 70.00);

insert into Servico (descricao, tipo, valor) values ('Ultrassom', 'exame', 150.00);

```
insert into Servico (descricao, tipo, valor) values ('Hemograma', 'exame', 60.00);
insert into Servico (descricao, tipo, valor) values ('V10', 'vacina', 120.00);
insert into Servico (descricao, tipo, valor) values ('Clinica geral', 'consulta', 80.00);
insert into Servico (descricao, tipo, valor) values ('Oftalmologia', 'consulta', 100.00);
```

```
insert into Funcionario (nome, salario, cpf, funcao) values ('Alberto Lima', 3500.00,
'12365478990', 'veterinario');
```

```
insert into Funcionario (nome, salario, cpf, funcao) values ('Luciano Silva', 3500.00,
'12365478770', 'veterinario');
```

```
insert into Funcionario (nome, salario, cpf, funcao) values ('Abel Braga', 1500.00,
'12365478770', 'assistente');
```

```
insert into Funcionario (nome, salario, cpf, funcao) values ('Luciana Barros', 1500.00,
'12365478770', 'secretaria');
```

```
insert into Funcionario (nome, salario, cpf, funcao) values ('Mariana Andrade', 3500.00,
'12365478773', 'veterinario');
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (2, 3, 1);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (6, 1, 2);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (3, 1, 3);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (7, 2, 4);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (3, 1, 5);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (1, 3, 6);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (2, 3, 6);
```

```
insert into Servico_Prestado (id_Servico, id_Funcionario, id_Atendimento) values (6, 5, 7);
```

```
insert into Veterinario (crm, especializacao, id_Funcionario) values ('123456789', 'clinico', 1);
```

```
insert into Veterinario (crm, especializacao, id_Funcionario) values ('123446789',
'oftalmologista', 2);
```

```
insert into Veterinario (crm, especializacao, id_Funcionario) values ('123446789', 'clinico', 5);
```

```
insert into Consulta (motivo, diagnostico, prescricao, id_ServicoPrestado) values ('Febre,
vomito', 'infeccao alimentar', 'amoxicilina', 2);
```

```
insert into Consulta (motivo, diagnostico, prescricao, id_ServicoPrestado) values ('Olho
esbranquiçado', 'glaucoma', 'tetraciclina', 4);
```

```
insert into Consulta (motivo, diagnostico, prescricao, id_ServicoPrestado) values ('Febre, vomito', 'infeccao alimentar', 'amoxicilina', 8);
```

```
update table servico set valor =  
case when tipo = 'vacina' then valor * 1.1  
when tipo = 'exame' then valor * 1.05  
when tipo = 'consulta' then valor * 1.15  
end;
```

1- Quantidade de clientes que possuem mais de um animal

```
select count(*) as 'clientes 2+ animais' from v;
```

create view v as

```
SELECT count(idAnimal) as qtde, id_Cliente from animal  
group by id_Cliente having qtde > 1 ;
```

2 - Quantas consultas cada veterinário fez

```
select count(idConsulta) as qtde_consultas, f.nome from consulta c  
inner join servico_prestado sp  
on c.id_ServicoPrestado = sp.idServicoPrestado  
inner join funcionario f  
on f.idFuncionario = sp.id_Funcionario  
group by f.nome;
```

3 - Médio do valor total dos atendimentos agrupados por espécie

```
select truncate(avg(total), 2) as 'media', especie from atendimento  
inner join animal  
on idAnimal = id_Animal  
group by especie;
```


4 - Quantos serviços foram realizados em um período por tipo

```
select count(idServico) as 'servicos', tipo from servico
inner join servico_prestado
on id_Servico = idServico
inner join atendimento
on id_Atendimento = idAtendimento
where data between '2015-06-20' and '2019-08-02'
group by tipo;
```

5 - Quantos clientes são de outra cidade

```
select count(idCliente) as 'qtde' from cliente
inner join endereco
on idEndereco = id_Endereco
where cidade <> 'campos dos goytacazes';
```

Comandos DCL

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY '1234';
```

```
GRANT ALL ON Clinica.* TO 'admin'@'localhost';
```

```
CREATE USER 'veterinario'@'localhost' IDENTIFIED BY '1234';
```

```
GRANT SELECT, UPDATE(prescricao, motivo, diagnostico) ON Clinica.Consulta TO
'veterinario'@'localhost';
```

```
GRANT SELECT ON Clinica.Animal TO 'veterinario'@'localhost';
```

```
GRANT SELECT ON Clinica.Cliente TO 'veterinario'@'localhost';
```

```
GRANT SELECT ON Clinica.Servico_Prestado TO 'veterinario'@'localhost';
```

```
CREATE USER 'secretario'@'localhost' IDENTIFIED BY '1234';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON Clinica.Endereco TO 'secretario'@'localhost';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON Clinica.Cliente TO 'secretario'@'localhost';
```

```

GRANT SELECT, INSERT, UPDATE, DELETE ON Clinica.Animal TO 'secretario'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON Clinica.Atendimento TO 'secretario'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON Clinica.Servico_Prestado TO
'secretario'@'localhost';

GRANT SELECT ON Clinica.Servico TO 'secretario'@'localhost';

GRANT SELECT(idFuncionario) ON Clinica.Funcionario TO 'secretario'@'localhost';

GRANT SELECT(idVeterinario) ON Clinica.Veterinario TO 'secretario'@'localhost';

GRANT SELECT(idConsulta, motivo), INSERT(idConsulta, motivo) ON Clinica.Consulta TO
'secretario'@'localhost';

```

Function, procedure e triggers

```

DELIMITER //

CREATE TRIGGER tgr_servico_prestado_bu

before UPDATE

ON Servico_Prestado

FOR EACH ROW BEGIN

IF NEW.concluido <> OLD.concluido THEN

    set @v1 := (select valor from servico inner join servico_prestado on idServico =
id_Servico where NEW.idServicoPrestado = idServicoPrestado);

    IF NEW.concluido = 1 THEN

        update atendimento set total = total + @v1 where NEW.id_Atendimento =
idAtendimento;

    ELSE

        update atendimento set total = total - @v1 where NEW.id_Atendimento =
idAtendimento;

    END IF;

END IF;

END; //

```

```
CREATE TRIGGER tgr_servico_prestado_bi BEFORE INSERT
ON servico_prestado
FOR EACH ROW BEGIN
    SET NEW.concluido = 0;
END; //
```

```
CREATE TRIGGER tgr_atendimento_bi BEFORE INSERT
ON atendimento
FOR EACH ROW BEGIN
    SET NEW.total = 0;
END; //
```

```
CREATE PROCEDURE obtem_idade(id INT)
BEGIN
    declare dias float;
    declare data_nasc date;
    declare idade float;
    select data_nascimento from animal where idAnimal= id into data_nasc;
    select datediff(curdate(), data_nasc) into dias;
    set idade = truncate((dias/365), 0);
    if idade < 1 then
        set idade = truncate((dias/30), 0);
        select concat(idade, ' meses') as 'idade';
    else
        select concat(idade, ' anos') as 'idade';
    end if;
END; //
```

```
CREATE FUNCTION calcula_bonus(inicio date, fim date, id int) RETURNS int DETERMINISTIC
BEGIN
    DECLARE bonus FLOAT;

    select sum(valor) from servico s inner join servico_prestado sp on idServico = id_Servico
    where tipo = 'consulta' and id_Funcionario = id into bonus;

    SET bonus = bonus * 0.1;

    RETURN bonus;

END; //
DELIMITER ;
```

Backup

O backup será feito diariamente às 19:30.

No arquivo bkp_clinica.bat:

```
cd C:\Users\clinica\ backup
mysqldump -u root -p1225 --databases Clinica > bkp_clinica.txt
```

Comando para agendar execução através do Agendador de Tarefas do Windows:

```
schtasks /Create /SC daily /TN "backup bd" /TR "C:\Users\clinica\backup\bkp_clinica.bat" /ST
19:30
```