

Projeto de Banco de Dados

Clínica Veterinária

Luiz Roberto
Lucas Florêncio

Banco de Dados - SI 23.1
Professor Robson Fidalgo



Centro de
Informática
UFPE

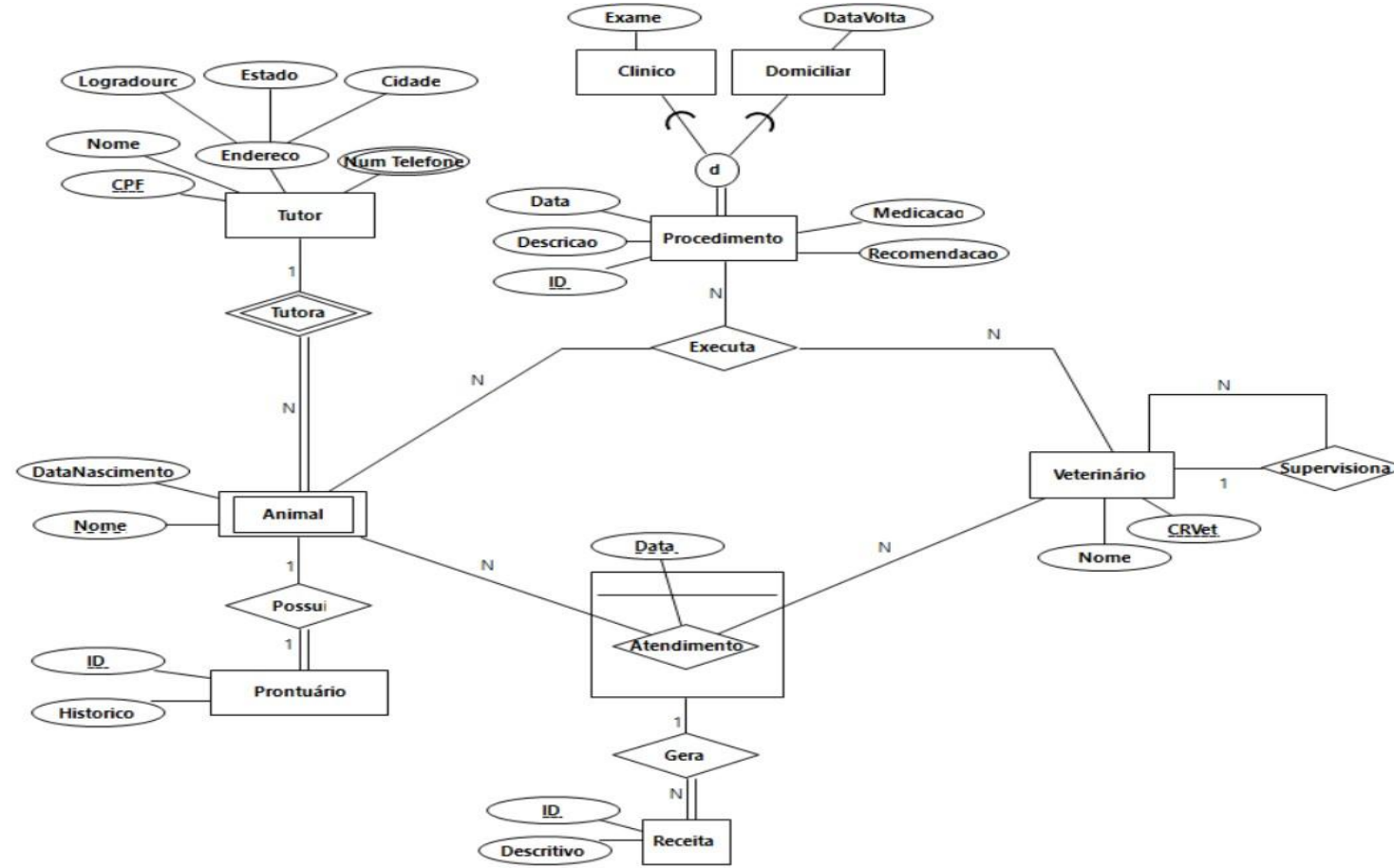


UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

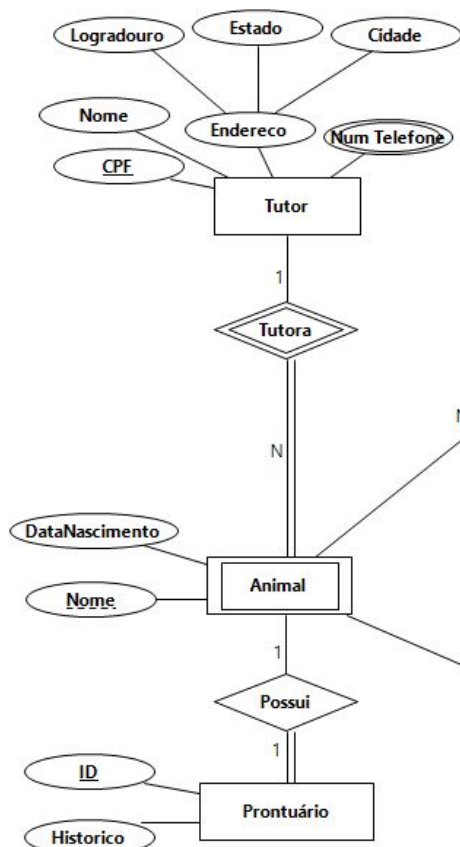
Contexto



Modelo Conceitual



Modelo Lógico



TUTOR (*CPF*, NOME, LOG, ESTADO, CIDADE)

NUM_TELEFONE (*CPF, NUMERO*)

CPF -> TUTOR(CPF)

ANIMAL (*NOME, CPF*, NASCIMENTO)

CPF -> TUTOR(CPF)

PRONTUARIO(*IDP*, HISTORICO, [CPF, NOME]!)

CPF, NOME -> ANIMAL(CPF, NOME)

Modelo Físico (DDL)

```
CREATE TABLE TUTOR (  
    CPF VARCHAR(11) PRIMARY KEY,  
    NOME VARCHAR(80) NOT NULL,  
    LOG VARCHAR(255),  
    ESTADO VARCHAR(50),  
    CIDADE VARCHAR(50)  
)
```

Modelo Físico (DDL)

```
CREATE TABLE NUM_TELEFONE (  
    CPF VARCHAR(11),  
    NUMERO VARCHAR(12),  
    PRIMARY KEY (CPF, NUMERO),  
    FOREIGN KEY (CPF) REFERENCES TUTOR(CPF)  
)
```

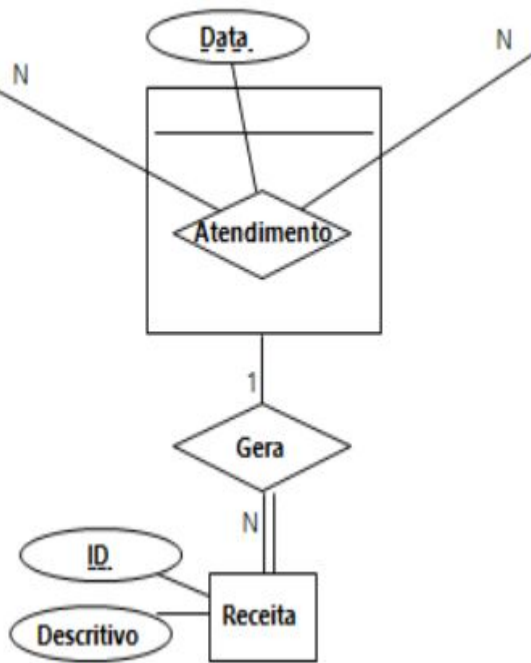
Modelo Físico (DDL)

```
CREATE TABLE ANIMAL (  
    CPF VARCHAR(11),  
    NASCIMENTO DATE,  
    NOME VARCHAR(80),  
    PRIMARY KEY (NOME, CPF),  
    FOREIGN KEY (CPF) REFERENCES TUTOR(CPF) ON  
    DELETE CASCADE  
)
```

Modelo Físico (DDL)

```
CREATE TABLE PRONTUARIO (  
    HISTORICO VARCHAR(4000),  
    IDP VARCHAR(4),  
    NOME VARCHAR(80) NOT NULL,  
    CPF VARCHAR(11) NOT NULL,  
    PRIMARY KEY (IDP),  
    FOREIGN KEY (NOME, CPF) REFERENCES  
    ANIMAL(NOME, CPF) ON DELETE CASCADE,  
    CONSTRAINT AK_ANIMAL UNIQUE (NOME, CPF)  
)
```


Modelo Lógico



ATENDIMENTO (DATA, NOME, CPF, CRVET)

NOME, CPF -> ANIMAL(NOME)

CRVET -> VETERINARIO(CRVET)

RECEITA (IDR, DESCRITIVO, DATA!, NOME!, CPF!, CRVET!)

DATA, NOME, CPF, CRVET -> ATENDIMENTO(DATA, NOME, CPF, CRVET)

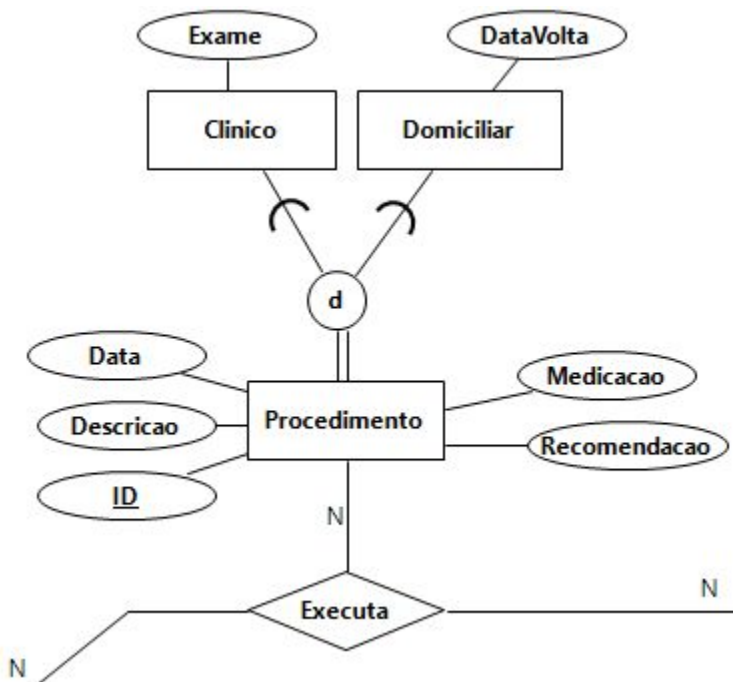
Modelo Físico (DDL)

```
CREATE TABLE ATENDIMENTO (  
    DATA DATE,  
    NOME VARCHAR(80),  
    CPF VARCHAR(11),  
    CRVET VARCHAR(10),  
    PRIMARY KEY (DATA, NOME, CPF, CRVET),  
    FOREIGN KEY (NOME, CPF) REFERENCES  
ANIMAL(NOME, CPF) ON DELETE CASCADE,  
    FOREIGN KEY (CRVET) REFERENCES  
VETERINARIO(CRVET) ON DELETE CASCADE  
)
```

Modelo Físico (DDL)

```
CREATE TABLE RECEITA (  
    IDR VARCHAR(4) PRIMARY KEY,  
    DESCRITIVO VARCHAR(255),  
    DATA DATE NOT NULL,  
    NOME VARCHAR(80) NOT NULL,  
    CPF VARCHAR(11) NOT NULL,  
    CRVET VARCHAR(10) NOT NULL,  
    FOREIGN KEY (DATA, NOME, CPF, CRVET) REFERENCES  
    ATENDIMENTO(DATA, NOME, CPF, CRVET)  
)
```

Modelo Lógico



PROCEDIMENTO(IDP, DESCRICAO, DATA, MEDICACAO, RECOMENDACAO)

CLINICO(IDP, EXAME)

IDP -> PROCEDIMENTO(IDP)

DOMICILIAR(IDP, DATAVOLTA)

IDP -> PROCEDIMENTO(IDP)

EXECUTA(NOME, CPF, IDP, CRVET)

NOME, CPF -> ANIMAL(NOME, CPF)

IDP -> PROCEDIMENTO(IDP)

CRVET -> VETERINARIO(CRVET)

Modelo Físico (DDL)

```
CREATE TABLE PROCEDIMENTO (  
    IDP VARCHAR(4) PRIMARY KEY,  
    DESCRICAO VARCHAR(255),  
    DATA DATE,  
    MEDICACAO VARCHAR(255),  
    RECOMENDACAO VARCHAR(255)  
)
```

Modelo Físico (DDL)

```
CREATE TABLE CLINICO (  
    IDP VARCHAR(4) PRIMARY KEY,  
    EXAME VARCHAR(255),  
    FOREIGN KEY (IDP) REFERENCES PROCEDIMENTO(IDP)  
ON DELETE CASCADE  
)
```

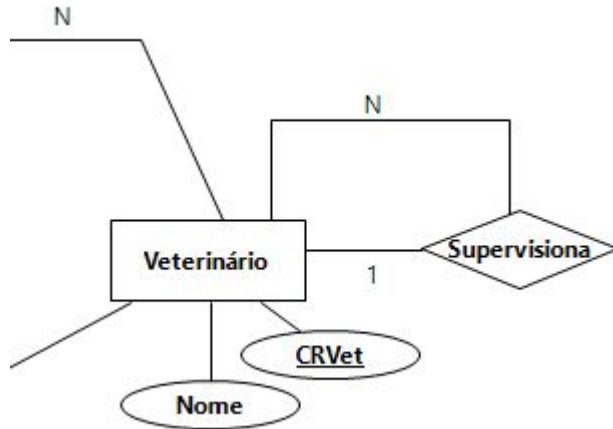
Modelo Físico (DDL)

```
CREATE TABLE DOMICILIAR (  
    IDP VARCHAR(4) PRIMARY KEY,  
    DATAVOLTA DATE,  
    FOREIGN KEY (IDP) REFERENCES PROCEDIMENTO(IDP)  
ON DELETE CASCADE  
)
```

Modelo Físico (DDL)

```
CREATE TABLE EXECUTA (  
    NOME VARCHAR(80),  
    CPF VARCHAR(11),  
    IDP VARCHAR(4),  
    CRVET VARCHAR(10),  
    PRIMARY KEY (NOME, CPF, IDP, CRVET),  
    FOREIGN KEY (NOME, CPF) REFERENCES ANIMAL(NOME, CPF) ON  
DELETE CASCADE,  
    FOREIGN KEY (IDP) REFERENCES PROCEDIMENTO(IDP) ON  
DELETE CASCADE,  
    FOREIGN KEY (CRVET) REFERENCES VETERINARIO(CRVET) ON  
DELETE CASCADE  
)
```


Modelo Lógico



VETERINARIO(CRVET, NOME, CRVETSUP)
CRVETSUP -> VETERINARIO(CRVET)

Modelo Físico (DDL)

```
CREATE TABLE VETERINARIO (  
    CRVET VARCHAR(10) PRIMARY KEY,  
    NOME VARCHAR(80) NOT NULL,  
    CRVETSUP VARCHAR(10),  
    FOREIGN KEY (CRVETSUP) REFERENCES  
    VETERINARIO(CRVET) ON DELETE SET NULL  
)
```

Modelo Físico (DML)

-- Consulta GROUP BY/HAVING, retorna nome dos tutores que possuem mais de um animal e a quantidade de animais desses.

```
SELECT T.NOME AS TUTOR, COUNT(A.NOME) AS  
QTD_ANIMAIS  
FROM TUTOR T  
JOIN ANIMAL A ON T.CPF = A.CPF  
GROUP BY T.NOME  
HAVING COUNT(A.NOME) > 1;
```

Modelo Físico (DML)

-- Consulta JUNÇÃO INTERNA, retorna os atendimentos armazenados no BD.

```
SELECT A.NOME AS ANIMAL, T.NOME AS TUTOR, V.NOME AS  
VETERINARIO, AT.DATA  
FROM ATENDIMENTO AT  
JOIN ANIMAL A ON AT.CPF = A.CPF  
JOIN TUTOR T ON A.CPF = T.CPF  
JOIN VETERINARIO V ON AT.CRVET = V.CRVET;
```

Modelo Físico (DML)

-- Consulta JUNÇÃO EXTERNA, retorna os atendimentos e as receitas prescritas nesses, incluindo os atendimentos sem receitas.

```
SELECT A.DATA, A.NOME AS NomeAnimal, A.CPF AS  
CPFTutor, A.CRVET AS CRVet, R.IDR, R.DESCRITIVO  
FROM ATENDIMENTO A  
LEFT JOIN RECEITA R ON A.DATA = R.DATA AND A.NOME =  
R.NOME AND A.CPF = R.CPF AND A.CRVET = R.CRVET  
ORDER BY A.DATA;
```

Modelo Físico (DML)

--Consulta SEMI JUNÇÃO, retorna os animais que já foram atendidos.

```
SELECT DISTINCT A.NOME, A.CPF
FROM ANIMAL A
WHERE EXISTS (
    SELECT 1
    FROM ATENDIMENTO AT
    WHERE A.NOME = AT.NOME AND A.CPF = AT.CPF
);
```

Modelo Físico (DML)

--Consulta ANTI JOIN, retorna os animais que não ainda não foram atendidos.

```
SELECT T.NOME AS TUTOR
FROM TUTOR T
WHERE NOT EXISTS (
    SELECT 1
    FROM ATENDIMENTO AT
    WHERE AT.CPF = T.CPF
);
```

Modelo Físico (DML)

-- Consulta com SUBCONSULTA DO TIPO ESCALAR, retorna os animais junto com o total de atendimentos que cada um teve.

```
SELECT A.NOME, A.CPF,  
       (SELECT COUNT(*)  
        FROM ATENDIMENTO AT  
        WHERE A.NOME = AT.NOME AND A.CPF = AT.CPF) AS  
TotalAtendimentos  
FROM ANIMAL A;
```


Modelo Físico (DML)

-- Consulta com SUBCONSULTA DO TIPO LINHA, retorna os nomes dos tutores que moram no mesmo estado e cidade do tutor com cpf 11111111111

```
SELECT NOME
FROM TUTOR
WHERE (CIDADE, ESTADO) =
      (SELECT CIDADE, ESTADO
       FROM TUTOR
       WHERE CPF = 11111111111);
```

Modelo Físico (DML)

-- Consulta com SUBCONSUTLA DO TIPO TABELA, retorna os animais juntamente com o total de receitas geradas para cada animal.

```
SELECT A.NOME, A.CPF, Q1.TotalReceitas
FROM ANIMAL A
INNER JOIN (
    SELECT NOME, CPF, COUNT(*) AS TotalReceitas
    FROM RECEITA
    GROUP BY NOME, CPF
) Q1 ON A.NOME = Q1.NOME AND A.CPF = Q1.CPF;
```

Modelo Físico (DML)

-- Consulta com OPERAÇÃO DE CONJUNTO, retorna informações detalhadas sobre os procedimentos realizados em animais, incluindo o nome do animal, o ID do procedimento, a descrição do procedimento, o tipo de procedimento e o nome do veterinário responsável.

Modelo Físico (DML)

```
SELECT
  A.NOME AS "Nome do Animal",
  E.IDP AS "ID do Procedimento",
  P.DESCRICAO AS "Descrição do Procedimento",
  'Domiciliar' AS "Tipo de Procedimento",
  V.NOME AS "Nome do Veterinário"
FROM
  EXECUTA E
INNER JOIN
  ANIMAL A ON E.NOME = A.NOME AND E.CPF = A.CPF
INNER JOIN
  PROCEDIMENTO P ON E.IDP = P.IDP
INNER JOIN
  VETERINARIO V ON E.CRVET = V.CRVET
INNER JOIN
  DOMICILIAR D ON E.IDP = D.IDP
```

Modelo Físico (DML)

UNION

SELECT

A.NOME AS "Nome do Animal",
E.IDP AS "ID do Procedimento",
P.DESCRICAO AS "Descrição do Procedimento",
'Clínico' AS "Tipo de Procedimento",
V.NOME AS "Nome do Veterinário"

FROM

EXECUTA E

INNER JOIN

ANIMAL A ON E.NOME = A.NOME AND E.CPF = A.CPF

INNER JOIN

PROCEDIMENTO P ON E.IDP = P.IDP

INNER JOIN

VETERINARIO V ON E.CRVET = V.CRVET

INNER JOIN

CLINICO C ON E.IDP = C.IDP;

Obrigado!

Luiz Roberto
Lucas Florêncio

13/09/2023