



Projeto Avaliativo Banco de Dados

Instruções

Para esta atividade você irá precisar da ajuda da ferramenta de IA [Chat GPT](#).

1. Você deverá gerar uma **modelagem conceitual** a partir do seguinte **prompt**:
 - a. “Gere a modelagem conceitual de um banco de dados inspirado na [plataforma].”;
 - b. A palavra usada para substituir [plataforma], deve se referenciar a alguma plataforma amplamente utilizada, tais como: Uber, Spotify, Facebook, Youtube, ou outro serviço amplamente utilizado. Caso não queira, especifique algum ambiente ou contexto: Biblioteca, Escola, Academia, Clínica Médica, Petshop, Futebol, etc;
 - c. A modelagem conceitual deverá ter no mínimo 4 tabelas e apresentar todas as cardinalidades (1:1, 1:N, N:N);
 - d. O link do chat usado para gerar a modelagem por meio do Chat GPT deve ser anexado neste documento.
2. Construa o **Diagrama Entidade-Relacionamento** a partir da descrição gerada pela IA fazendo os ajustes necessários em nomenclatura, estrutura e etc. Faça uso do software online [brModelo](#).
 - a. Você deverá anexar um print em boa qualidade do diagrama em questão.
3. Construa a **modelagem física** por meio da linguagem **SQL**, o código deve ser feito manualmente baseado no Diagrama Entidade-Relacionamento construído a partir do rascunho proposto pela IA e adaptado por você.
 - a. Você deverá entregar um único arquivo SQL com todos os códigos que forem solicitados.
4. Aplique comandos diversos da **DML** e **DQL**. Tais como: *INSERT*, *UPDATE*, *DELETE*, *SELECT*. Aplique filtragens diversas envolvendo tudo que foi estudado ao longo do módulo: junção, agregação e etc.

Questionário

Responda atentamente às seguintes questões propostas a seguir:

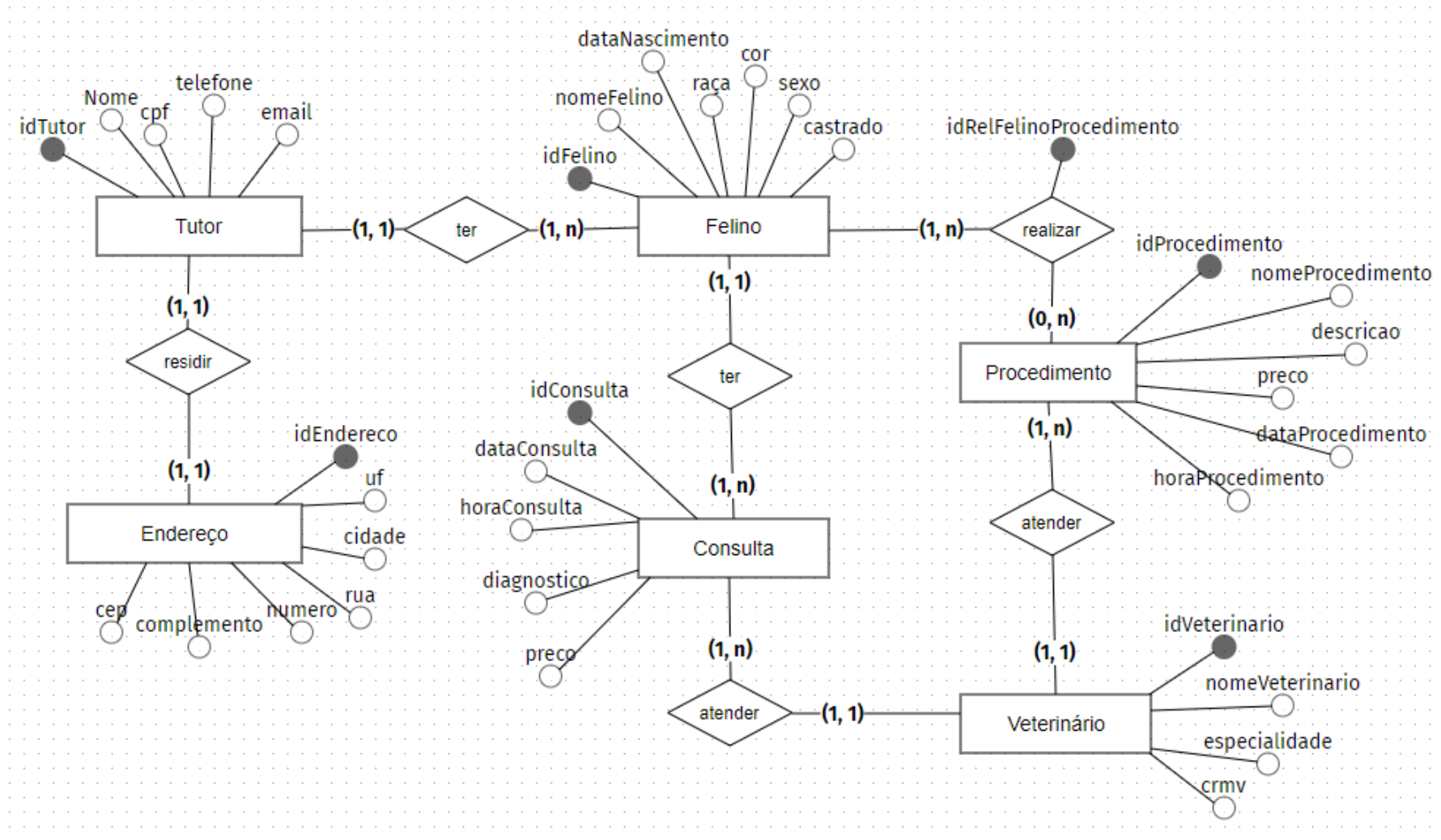
1) Indique o link do chat compartilhado do Chat GPT.

<https://chat.openai.com/share/fef0e5ab-8c82-4a12-96dc-8456cc8a97af>

2) Descreva brevemente suas adaptações a respeito da modelagem proposta pela IA.

- A IA gerou 5 entidades, alterei o nome de 2 (de 'Dono' para 'Tutor' e de 'Gato' para 'Felino') e acrescentei mais 1 entidade, no caso, tirei o endereço da entidade 'Tutor' e criei a entidade 'Endereço'.
- Exclui alguns atributos sugeridos.
- Incluí alguns atributos.
- Separei o relacionamento das entidades 'Consulta' e 'Procedimento'. Coloquei ambas ligadas à entidade 'Felino'.
- Criei uma tabela de relacionamento entre as entidades 'Felino' e 'Procedimento'.

3) Anexe abaixo a imagem do Diagrama Entidade-Relacionamento construído com o brModelo.



4) O banco de dados está normalizado conforme 1FN, 2FN e 3FN? Justifique.

Sim, está normalizado em 1FN, 2FN e 3FN.

- 1FN, porque não tem repetições de dados em nenhuma das tabelas;
- 2FN, porque não tem dependências transitivas;
- 3FN, porque não tem dependências funcionais sobre as chaves candidatas transitivas.

5) Imagine como poderia ser o seu banco de dados em um modelo não-relacional. Leia sobre [Data Modelling](#), [Model One-to-One](#) e [One-to-Many](#) no MongoDB.

O MongoDB usa coleções de documentos, que pelo que entendi são mais flexíveis e escaláveis do que as tabelas. Ele também usa relações para conectar os diferentes documentos, o que permite que os dados sejam relacionados de forma lógica, também.

Acredito que seria muito boa sua utilização neste projeto da 'Clínica veterinária de felinos'.

No geral, tudo vai depender do projeto que será utilizado, por isso é importante entendermos bem o projeto e entendermos bem as ferramentas, para podermos tomar a melhor decisão, ainda mais quando se trata de dados.

6) Cole na área abaixo o código SQL usado para a modelagem física.

```
CREATE DATABASE clinicaVeterinaria_db;  
USE clinicaVeterinaria_db;
```

```
CREATE TABLE tutor(  
    idTutor INTEGER PRIMARY KEY AUTO_INCREMENT,  
    nomeTutor VARCHAR(100) NOT NULL,  
    cpf VARCHAR(11) UNIQUE NOT NULL,  
    telefone VARCHAR(11) NOT NULL,  
    email VARCHAR(50)  
);
```

```
CREATE TABLE endereco(  
    idEndereco INTEGER PRIMARY KEY AUTO_INCREMENT,  
    uf VARCHAR(2) NOT NULL,  
    cidade VARCHAR(40) NOT NULL,  
    rua VARCHAR(100) NOT NULL,  
    numero VARCHAR(10) NOT NULL,  
    complemento VARCHAR(40) DEFAULT("Sem complemento"),  
    cep VARCHAR(8) NOT NULL,  
    fkTutor INTEGER UNIQUE NOT NULL,  
    FOREIGN KEY(fkTutor) REFERENCES tutor(idTutor)  
);
```

```
CREATE TABLE felino(  
    idFelino INTEGER PRIMARY KEY AUTO_INCREMENT,  
    nomeFelino VARCHAR(30) NOT NULL,  
    dataNascimento DATE NOT NULL,  
    raca VARCHAR(25) NOT NULL DEFAULT("Sem raça definida"),  
    cor VARCHAR(20) NOT NULL,  
    sexo VARCHAR(5) NOT NULL,  
    castrado VARCHAR(3) NOT NULL,  
    fkTutor INTEGER NOT NULL,  
    FOREIGN KEY(fkTutor) REFERENCES tutor(idTutor)  
);
```

```
CREATE TABLE veterinario(  
    idVeterinario INTEGER PRIMARY KEY AUTO_INCREMENT,
```

```

nomeVeterinario VARCHAR(100) NOT NULL,
especialidade VARCHAR(30) NOT NULL,
crmV VARCHAR(10) UNIQUE NOT NULL
);

CREATE TABLE consulta(
  idConsulta INTEGER PRIMARY KEY AUTO_INCREMENT,
  dataConsulta DATE NOT NULL,
  horaConsulta TIME NOT NULL,
  preco DECIMAL(5,2) NOT NULL DEFAULT(100.00),
  diagnostico VARCHAR(150) NOT NULL,
  fkFelino INTEGER NOT NULL,
  fkVeterinario INTEGER NOT NULL,
  FOREIGN KEY(fkFelino) REFERENCES felino(idFelino),
  FOREIGN KEY(fkVeterinario) REFERENCES veterinario(idVeterinario)
);

CREATE TABLE procedimento(
  idProcedimento INTEGER PRIMARY KEY AUTO_INCREMENT,
  nomeProcedimento VARCHAR(50) NOT NULL,
  Descricao VARCHAR(100) NOT NULL,
  preco DECIMAL(7, 2) NOT NULL,
  fkVeterinario INTEGER NOT NULL,
  FOREIGN KEY(fkVeterinario) REFERENCES veterinario(idVeterinario)
);

CREATE TABLE rel_felino_procedimento(
  idRelFelinoProcedimento INTEGER PRIMARY KEY AUTO_INCREMENT,
  dataProcedimento DATE NOT NULL,
  horaProcedimento TIME NOT NULL,
  fkFelino INTEGER NOT NULL,
  fkProcedimento INTEGER NOT NULL,
  FOREIGN KEY(fkFelino) REFERENCES felino(idFelino),
  FOREIGN KEY(fkProcedimento) REFERENCES procedimento(idProcedimento)
);

```

7) Cite algumas filtragens que realizou e o que podemos interpretar delas para gerar informação útil de um banco de dados.

No geral, os filtros ajudam bastante na gestão da clínica, tanto na questão financeira quanto na produtividade. Pois é possível entender quantas consultas e procedimentos estão sendo realizados por veterinário, qual é o valor médio dos procedimentos. Além disso, é possível entender qual a frequência que cada paciente vai ao veterinário e sua faixa de idade, ajudando a criar pacotes de serviços que atendam a necessidade de cada paciente. Essas são algumas das informações úteis que os filtros podem nos ajudar.

-- Aumento do procedimento de Castração de R\$ 500.00 para R\$ 600.00

UPDATE procedimento

SET preco = preco * 1.2

WHERE nomeProcedimento = 'Castração';

-- Quantidade Total de procedimentos agrupados por veterinário

SELECT COUNT(*) AS QtdProcedimentos, vet.nomeVeterinario

FROM rel_felino_procedimento AS rel

JOIN procedimento AS proc

ON proc.idProcedimento = rel.fkProcedimento

JOIN veterinario AS vet

ON vet.idVeterinario = proc.fkVeterinario

GROUP BY vet.idVeterinario;

-- Liste os nomes dos pacientes atendidos em consulta pelo veterinário com o id = 4

SELECT fel.nomeFelino

FROM felino AS fel

JOIN consulta AS cons

ON fel.idFelino = cons.fkFelino

WHERE cons.fkVeterinario = 4;

-- Informe o valor total recebido, em procedimentos, pelo veterinario com o id = 1

```
SELECT SUM(proc.preco) AS valorRecebido
FROM procedimento AS proc
      JOIN rel_felino_procedimento AS rel
      ON proc.idProcedimento = rel.fkProcedimento
WHERE proc.fkVeterinario = 1;

-- Mostra a média dos preços dos procedimentos.
SELECT AVG(preco) AS mediaProcedimento
FROM procedimento;

-- Mostra os valores mínimo e máximo dos procedimentos.
SELECT MIN(preco) AS menorPreco, MAX(preco) AS maiorPreco
FROM procedimento;

-- Mostra os 5 gatos que nasceram mais próximos da data de hoje.
SELECT *
FROM felino
      ORDER BY ABS(DATEDIFF(dataNascimento, CURRENT_DATE()))
      LIMIT 5;
```

Observação: Cuidado para não fugir do padrão de formatação proposto para o documento.