TRABALHO / RESOLUÇÃO:

1)

a) $\Pi_{Modelo,Velocidade}(Dados)$

O operador de projeção (Π), é utilizado para selecionar determinadas colunas de uma relação. A operação é executada em apenas uma relação e o resultado é uma nova relação contendo apenas os atributos selecionados, eliminando-se as duplicidades.

b) $\Pi_{Carros.matricula,\ Carros.modelo,\ Dados.combustivel}$

$$\sigma_{Dados.velocidade>310}(Carros \bowtie_{Carros.modelo=Dados.modelo} Dados)$$

2)

a) CREATE ROLE Responsavel

b) GRANT Responsavel
TO ‘Ana’ , ‘Bruno’

c) GRANT SELECT
ON Carros
TO Responsavel

d) REVOKE SELECT
ON Carros
FROM Responsavel

3) Os dados estão na Primeira Forma Normal (1FN) porque não existem elementos repetidos e nem atributos a ter mais de um valor.

Na Segunda Forma Normal (2FN) os atributos normais devem depender unicamente da chave primária da tabela. As colunas da tabela que não são dependentes dessa chave principal devem ser removidas e criar-se uma nova tabela utilizando esses dados. Assim temos:

Numero_cliente -> nome_cliente

Matricula -> modelo, responsável, telefone_responsavel

Numero_cliente, Matricula -> dias_aluguer

Na Terceira Forma Normal (3FN) todos os atributos de uma tabela devem ser funcionalmente independentes uns dos outros, ao mesmo tempo que devem ser exclusivamente independentes da chave primária. Assim temos:

Numero_cliente -> nome_cliente

Matricula -> modelo, responsável

Responsavel -> telefone_responsavel

Numero_cliente, Matricula -> dias_aluguer

4)

a) Entidades:

livro(id_livro, titulo, ISBN, num_paginas, ano_edicao)

autor(nome, biografia, período_atividade, genero_literario)

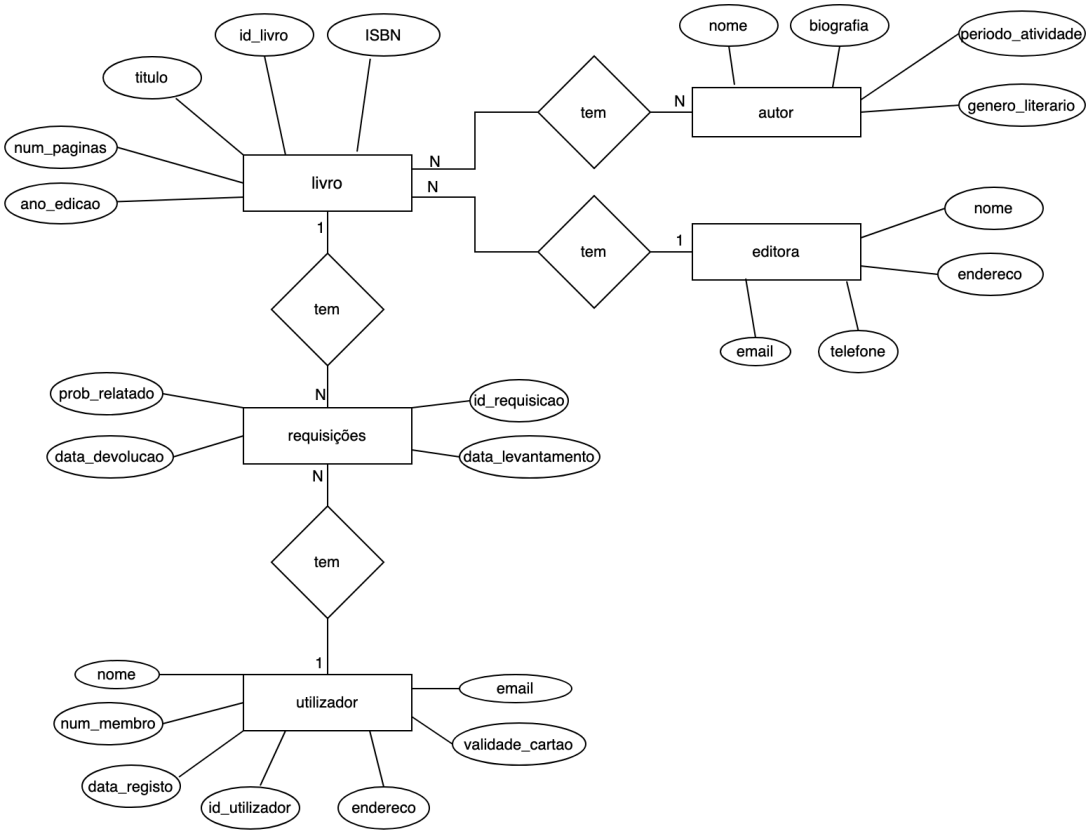
editora(nome, endereço, telefone, email)

requisições(id_requisição, data_levantamento, data_devolucao, prob_relatado)

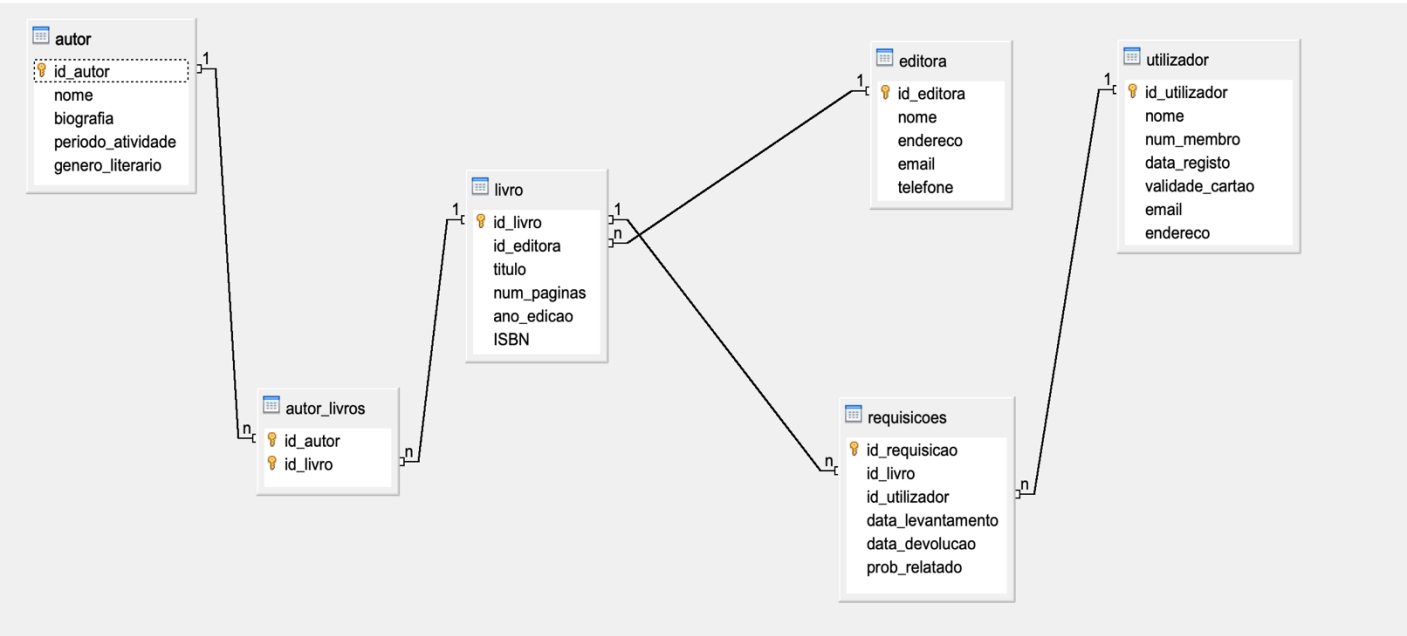
utilizador(id_utilizador, nome, num_membro, validade_cartao, endereco, data_registo, email)

Relacionamentos:

- livro – autor, tem, N:N, cada livro pode ter um ou vários autores e cada autor pode ter escrito um ou vários livros
- editora – livro, tem, 1:N, cada editora pode ter um ou vários livros
- livro – requisições, tem, 1:N, cada livro pode ter um ou várias requisições e cada requisição pode ter um livro (como o enunciado refere “para cada requisição de um livro”, parti do pressuposto que tem de haver uma requisição por cada livro, caso contrário, se cada requisição pudesse ter vários livros então seria uma relação de muitos para muitos)
- utilizador – requisições, tem, 1:N, cada utilizador pode ter uma ou várias requisições


id_livro
ISBN
titulo
num_paginas
ano_edicao
livro
N
N
1
tem
N
autor
nome
biografia
periodo_atividade
genero_literario
tem
1
editora
nome
endereco
email
telefone
tem
N
requisições
prob_relatado
data_devolucao
id_requisicao
data_levantamento
N
tem
1
utilizador
nome
num_membro
data_registo
id_utilizador
endereco
email
validade_cartao

b)


autor
id_autor
nome
biografia
periodo_atividade
genero_literario
livro
id_livro
id_editora
titulo
num_paginas
ano_edicao
ISBN
editora
id_editora
nome
endereco
email
telefone
utilizador
id_utilizador
nome
num_membro
data_registo
validade_cartao
email
endereco
autor_livros
id_autor
id_livro
requisicoes
id_requisicao
id_livro
id_utilizador
data_levantamento
data_devolucao
prob_relatado