



Universidade Federal de Sergipe - UFS
Centro de Ciências Exatas e Tecnologia - CCET
Departamento de Computação - DCOMP
COMP0455 - Banco de Dados I (2019.1 - T02)
Docente: André Britto de Carvalho
Discente: Tiago Alves de Farias

Trabalho Prático de Banco de Dados

Sistema de Gerenciamento para Condomínios

São Cristóvão - SE

18 de Junho de 2019

Sumário

1. Introdução	3
1.1 Escopo	3
1.2 Funções do Sistema	3
2. Descrição do Banco de Dados	4
2.1 Requisitos	4
3. Projeto Conceitual	6
3.1 Modelo Entidade Relacionamento	6
4. Projeto Lógico	6
4.1 Regras para o Mapeamento	7
4.2 Descrição do Mapeamento / Esquema Relacional	7
4.3 Modelo Relacional	11
Referências	12

1. Introdução

1.1 Escopo

Este documento tem como objetivo descrever de forma abstrata um sistema de gerenciamento para condomínios. Tendo como principal característica o controle/monitoramento de moradores. Além disso, o gerenciamento dos funcionários.

1.2 Funções do Sistema

O Sistema de Gerenciamento para Condomínios possui as seguintes funções:

- Controle de moradores e funcionários através da inserção, consulta, alteração e remoção dos mesmos;
- Consultar os moradores que residem em um determinado apartamento;
- Consultar todos moradores que residem no condomínio;
- Consultar todos funcionários do condomínio;
- Verificar quais apartamentos estão disponíveis.

2. Descrição do Banco de Dados

2.1 Requisitos

REQ01: Uma pessoa é definida por seu primeiro nome, sobrenome, telefone, data de nascimento, data de entrada, data de saída, idade e o CPF que identifica cada pessoa unicamente.

REQ02: Nem toda entidade *Pessoa* possui uma entidade correspondente em *Morador* ou *Funcionário*.

REQ03: Um(a) morador(a) é definido por seu primeiro nome, sobrenome, telefone, data de nascimento, idade, número do bloco, número do apartamento, data de entrada, data de saída, e-mail e o CPF que identifica cada morador(a) unicamente.

REQ04: Um apartamento é definido por seu número.

REQ05: Cada morador(a) pode morar apenas em um apartamento.

REQ06: Cada apartamento possui nenhum ou vários moradores.

REQ07: Uma conta é definida por id_cont, IPTU, energia, água e a taxa do condomínio.

REQ08: Cada apartamento possui uma conta.

REQ09: Cada conta está vinculada à apenas um apartamento.

REQ10: Um bloco é definido por seu número.

REQ11: Cada apartamento está vinculado à apenas um bloco.

REQ12: Cada bloco possui vários apartamentos.

REQ13: Através do atributo Status, deve ser possível verificar se um determinado apartamento está ou não disponível.

REQ14: Um condomínio é definido por seu nome, id_cond, CEP, bairro, rua e o número da rua.

REQ15: Cada bloco possui um condomínio.

REQ16: Cada condomínio possui um ou vários blocos.

REQ17: O endereço é definido por CEP, bairro, rua e o número da rua.

REQ18: Um(a) funcionário(a) é definido por seu primeiro nome, sobrenome, telefone, data de nascimento, idade, cargo, salário, data de entrada, data de saída, CEP, bairro, rua, número da rua e o CPF que identifica cada funcionário(a) unicamente.

REQ19: Nem toda entidade *Funcionário* possui uma entidade correspondente em *Porteiro*, *Zelador* ou *Vigilante*.

REQ20: Cada zelador limpa nenhum ou vários blocos.

REQ21: Cada bloco é limpo por nenhum ou vários zeladores.

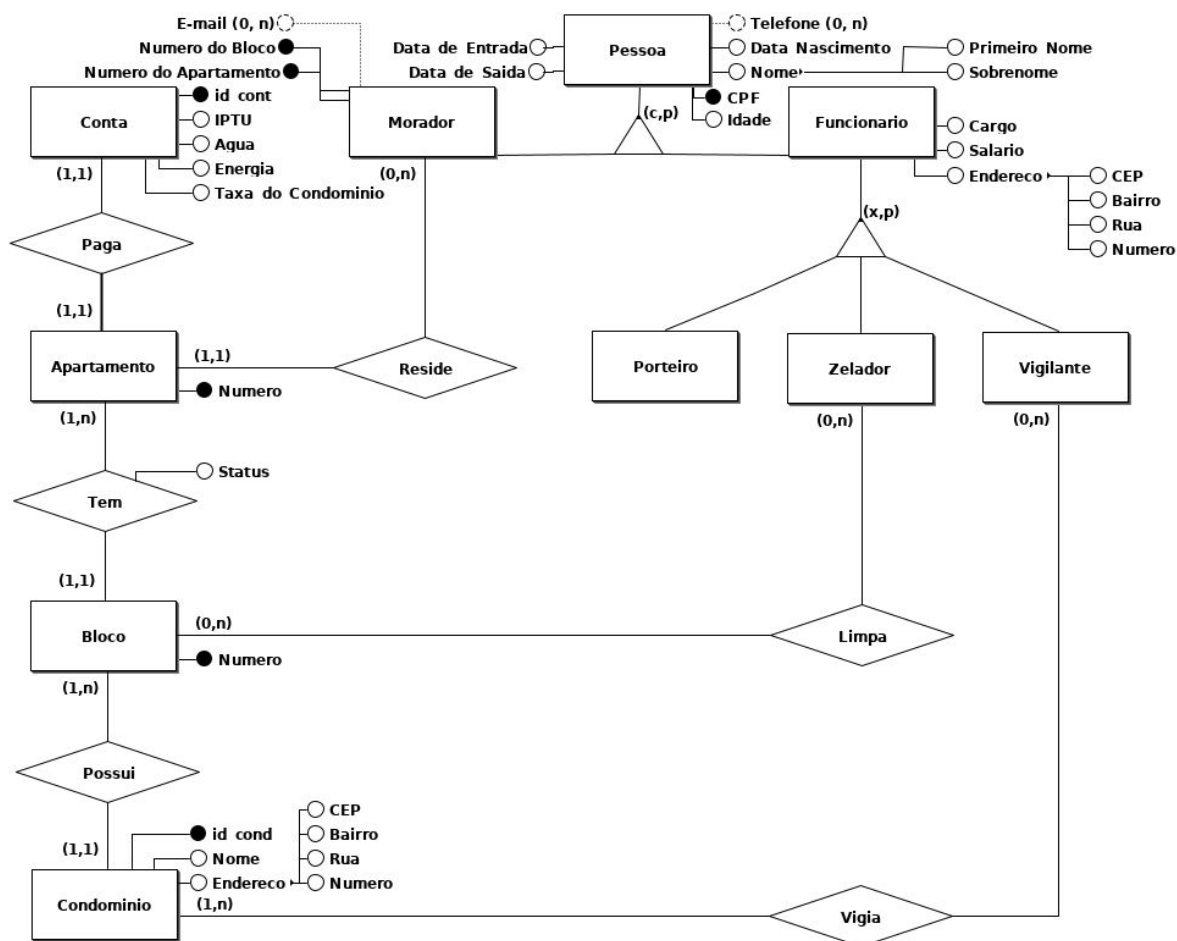
REQ22: Cada vigilante vigia um ou vários condomínios.

REQ23: Cada condomínio possui nenhum ou vários vigilantes.

REQ24: O banco de dados deve ser capaz de guardar informações sobre os moradores existentes em cada apartamento e os funcionários que trabalham no condomínio.

3. Projeto Conceitual

3.1 Modelo Entidade Relacionamento



4. Projeto Lógico

4.1 Regras para o Mapeamento

1. Toda entidade vira uma tabela.
2. Se o atributo for obrigatório, coloca-se NN (Not Null).
3. Atributo identificador (id): se torna chave primária (PK).
4. Atributos derivados: não modela.
5. Atributos compostos: tornam-se atributos simples.
6. Atributos multi-valorados: deve-se criar uma tabela, com o atributo e um id, também criar uma chave estrangeira (FK) do id para a tabela original.
7. Relacionamentos um para um (1:1): cria-se um atributo em uma tabela e relaciona a outra tabela.
8. Relacionamentos um para muitos (1:n): cria-se um atributo do lado muitos e relaciona a tabela um.
9. Relacionamentos muitos para muitos (n:m): deve-se criar uma tabela com chaves estrangeiras apontando para as chaves primárias das outras tabelas.
10. Relacionamentos ternários: obrigatoriamente cria-se uma tabela.
11. Entidades fracas: a FK obrigatoriamente é PK.
12. Generalização/Especialização: a PK da tabela filho relaciona a PK da tabela pai.

4.2 Descrição do Mapeamento / Esquema Relacional

Tabela Pessoa

- O atributo derivado não foi modelado (idade).
- Os atributos compostos passou a ser simples (primeiro_nome, sobrenome).
- O atributo multi-valorado telefone, passou à ser simples (telefone_celular).
- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.

Esquema relacional correspondente:

Pessoa (CPE, primeiro_nome NN, sobrenome NN, data_de_nascimento NN, telefone_celular, data_de_entrada NN, data_de_saida)

Tabela Morador

- O atributo multi-valorado e-mail, passou à ser simples (email).
- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.

Esquema relacional correspondente:

Morador (Pessoa_CPF, numero_do_bloco NN, numero_do_apartamento NN, email, Apartamento_id NN)

FK: (Pessoa_CPF) -> Pessoa(CPF)

FK: (Apartamento_id) -> Apartamento(id)

Tabela Apartamento

- O atributo sublinhado é o id.

Esquema relacional correspondente:

Apartamento (id)

Tabela Conta

- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.

Esquema relacional correspondente:

Conta (Apartamento_id, IPTU, agua, energia, taxa_do_condominio NN)

FK: (Apartamento_id) -> Apartamento(id)

Tabela Tem

- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.
- A tabela Tem era um relacionamento que passou à ser uma tabela.

Esquema relacional correspondente:

Tem (id, status, Apartamento_id NN, Bloco_id NN)

FK: (Apartamento_id) -> Apartamento(id)

(Bloco_id) -> Bloco(id)

Tabela Bloco

- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.

Esquema relacional correspondente:

Bloco (id, Condominio_id NN)

FK: (Condominio_id) -> Condominio(id)

Tabela Condominio

- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- O atributo numero passou a ser nomeado como numero_da_ rua.
- Os atributos compostos passou a ser simples (CEP, bairro, rua, numero_da_ rua).

Esquema relacional correspondente:

Condominio (id, nome NN, CEP NN, bairro NN, rua NN, numero_da_ rua NN)

Tabela Funcionario

- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.
- O atributo numero passou a ser nomeado como numero_da_ rua.
- Os atributos compostos passou a ser simples (CEP, bairro, rua, numero_da_ rua).

Esquema relacional correspondente:

Funcionario (Pessoa_CPF, cargo NN, salario NN, CEP NN, bairro NN, rua NN, numero_da_ rua NN)

FK: (Pessoa_CPF) -> Pessoa(CPF)

Tabela Porteiro

- O atributo sublinhado é o id.
- Os atributos FK são chaves estrangeiras.

Esquema relacional correspondente:

Porteiro (Funcionario_Pessoa_CPF)

FK: (Funcionario_Pessoa_CPF) -> Funcionario(Pessoa_CPF)

Tabela Zelador

- O atributo sublinhado é o id.
- Os atributos FK são chaves estrangeiras.

Esquema relacional correspondente:

Zelador (Funcionario_Pessoa_CPF)

FK: (Funcionario_Pessoa_CPF) -> Funcionario(Pessoa_CPF)

Tabela Limpa

- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.
- A tabela Limpa era um relacionamento n:m que virou uma tabela para relacionar as tabelas Zelador e Bloco.

Esquema relacional correspondente:

Limpa (Zelador_Funcionario_Pessoa_CPF NN, Bloco_id NN)

FK: (Zelador_Funcionario_Pessoa_CPF) -> Zelador(Funcionario_Pessoa_CPF)
(Bloco_id) -> Bloco(id)

Tabela Vigilante

- O atributo sublinhado é o id.
- Os atributos FK são chaves estrangeiras.

Esquema relacional correspondente:

Vigilante (Funcionario_Pessoa_CPF)

FK: (Funcionario_Pessoa_CPF) -> Funcionario(Pessoa_CPF)

Tabela Vigia

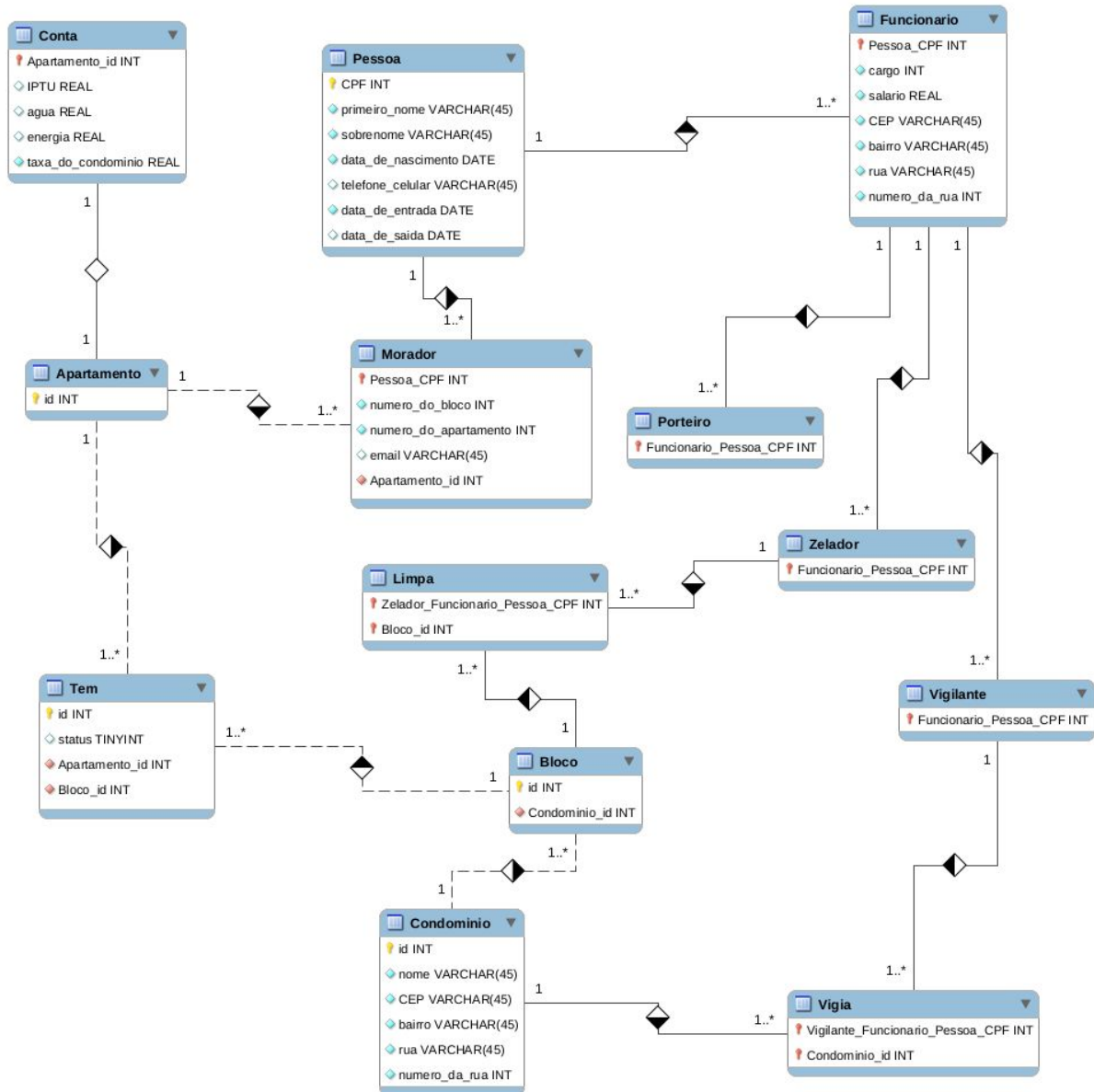
- O atributo sublinhado é o id.
- Os atributos obrigatórios possuem NN.
- Os atributos FK são chaves estrangeiras.
- A tabela Vigia era um relacionamento n:m que virou uma tabela para relacionar as tabelas Vigilante e Condominio.

Esquema relacional correspondente:

Vigia (Vigilante_Funcionario_Pessoa_CPF NN, Condominio_id NN)

FK: (Vigilante_Funcionario_Pessoa_CPF) -> Vigilante(Funcionario_Pessoa_CPF)
(Condominio_id) -> Condominio(id)

4.3 Modelo Relacional



Referências

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 5. ed. Porto Alegre, RS: Instituto de Informática da UFRGS: Sagra, 2004. 236 p.

ELMASRI, Ramez; NAVATHE, Sham. **Sistemas de banco de dados**. 6. ed. São Paulo, SP: Pearson, 2011. 788 p.

SILBERSCHATZ, Abraham; KORTH, Henry F; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro, RJ: Elsevier, 2006. 781 p.