

UNIVERSIDADE ESTADUAL PAULISTA

"JÚLIO DE MESQUITA FILHO"

Câmpus de Presidente Prudente

DIRETORIA TÉCNICA DE BIBLIOTECA E
DOCUMENTAÇÃO

Bacharelado em Ciência da Computação

Projeto Semestral da Disciplina de Banco de Dados

Sistema de gerenciamento ESF MULTI MISTURA

Professor: Ronaldo Celso Messias Correia

Aluno: André Luiz dos Santos Sanches

Data de entrega: 12/07/2021

Sumário

Especificação do problema	3
Esquema conceitual.....	4
Esquema relacional.....	5
Normalização.....	6

1 Especificação do Problema

Um bom sistema de saúde é a base para uma sociedade saudável, porém para melhor atender a população, a saúde carece de sistemas que auxiliem os atendimentos, com isso este trabalho será sobre como o sistema para atendimentos de postos de saúde municipais ajudam e agilizam tais atendimentos.

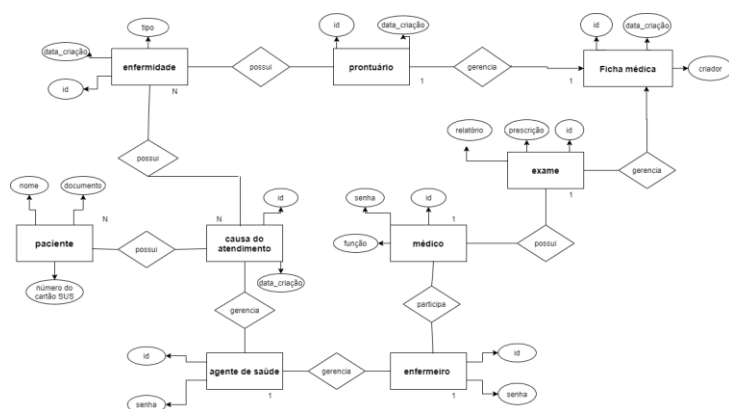
O sistema de gerenciamento de ESF Multi mistura é o responsável por salvar os dados médicos dos pacientes a cada consulta feita em postos de saúde municipais, salvando como documento o histórico médico dos pacientes, esses históricos só podem ser acessados por pessoas específicas, esta ficha de atendimento possui as seguintes entidades:

1. Paciente;
2. Médico;
3. Enfermeiro;
4. Agente de Saúde;
5. Causa do atendimento;
6. Enfermidade;
7. Exame;
8. Ficha médica;
9. Prontuário;

O paciente tem como atributo nome, documento e número do cartão SUS para acessar sua ficha médica, o médico tem como especialização sua função, que poderá ser: Fonólogo, psicólogo, psiquiatra, fisioterapeuta e dentista e seus atributos serão: senha, id e

O paciente possui N enfermidades que tem como atributos id, tipo e data_criação. Após feito o prontuário do paciente baseado em sua enfermidade, o prontuário terá como atributos id e data_criação; A causa do atendimento tem como atributos id e data_criação; Ambos agente de saúde e enfermeiro possuem como atributos id e senha.

2 Esquema conceitual



3 Esquema Relacional

1. Entidades:

- Paciente (**nome**, documento, número do cartão SUS)
- Agente de saúde (senha, **id**)
- Enfermeiro (senha, **id**)
- Médico (função, senha, **id**)

2. Entidades fracas;

3. Cardinalidade 1:1

- PossuiMédico(id, função, prescrição)
- GerenciaProntuário(id, data_criação)

4. Cardinalidade 1:N

- PossuiEnfermidade(id, tipo, data_criação)

5. Cardinalidade N:N

- PossuiPaciente (idPaciente*, idCausaAtendimento*)
- PossuiEnfermidade (idPaciente*, idEnfermidade*)

6. Relacionamentos n-ários;

7. Atributos Multivalorados;

8. Especializações;

- Paciente (nome, **número do cartão SUS**)
- Agente de saúde (id, **senha**)
- Enfermeiro (id, **senha**)
- Médico (id, **senha**)

4 Normalização

Não foi necessário normalizar as relações pois elas obedecem as 3 normas formais.

1ª Forma Normal: todos os atributos das relações são atômicos e monovalorados.

2ª Forma Normal: está na 1a.FN e não existe atributo não chave que é dependente de somente uma parte da chave primária.

3ª Forma Normal: está na 2a.FN e não existem atributos não chave que sejam dependentes de outros atributos não chave.

5 Especificação de Consultas em Álgebra Relacional e SQL

1. Todas as enfermidades registradas em 2021

$$\pi (id, data_criacao, tipo) (\sigma (data_criacao > 2021 - 01 - 01 \wedge data_criacao < 2022 - 01 - 01) (PossuiEnfermidade))$$

A equação exibe todas as enfermidades registradas no ano de 2021, em SQL temos o seguinte código:

```

1 SELECT id , criacao , tipo FROM PossuiEnfermidade WHERE criacao
2 BETWEEN ' 2021-01-01 ' AND ' 2022-01-01 ' ;

```

2. Todos os pacientes que realizaram consulta e as suas enfermidades:

$$\pi (\text{email}, \text{enfermidade.nome}, \text{estado}) (\sigma (\text{estado} = \text{"ok"}) (\text{paciente} ./ (\text{nome} = \text{nome})$$

$$(\text{possuiEnfermidade} ./ (\text{id} = \text{idCausaAtendimento}) \text{possuiPaciente}))$$

A equação exibe a consulta em álgebra relacional, em SQL temos o seguinte código:

```

1 SELECT paciente . nome, enfermidade. nome, enfermidade. estado
   FROM exame
2 INNER JOIN possuiPaciente ON exame. id=idPaciente .
   idCausaAtendimento
3 INNER JOIN possuiEnfermidade ON possuiEnfermidade.
   Id=idEnfermidade
4 INNER JOIN usuario ON usuario. email=possuiEnfermidade . Email
5 WHERE enfermidade. Estado =" ok " ;

```

3. Todos os relatórios das enfermidades:

$$\pi (\text{enfermidade.nome}) (\sigma (\text{tipo} = \text{"tipo1"}) (\text{enfermidade} ./ (\text{id} = \text{idEnfermidade}) (\text{Prontuário}$$

$$./ (\text{Prontuário.médico} = \text{possuiProntuário}) \text{possuiProntuário}))$$

A equação exibe esta consulta em álgebra relacional. Em SQL temos o seguinte código:

```

1 SELECT materia . nome FROM fonte INNER JOIN possuiProntuário
2 ON Prontuário. médico=possuiProntuário. médico INNER JOIN
   enfermidade
3 ON id=idEnfermidade WHERE prontuário . médico=" tipo1 " ;

```

4. Todos os médicos que realizaram exame em 2021:

$$\pi(\text{email}, \text{idExame})(\sigma(\text{atendimento}=2021)(\text{médico} \bowtie (\text{email}=\text{criador}) \\ \text{exame}))$$

A equação exibe esta consulta em álgebra relacional. Em SQL temos o seguinte código:

```

1 SELECT email FROM médico INNER JOIN exame
2 ON email=criador WHERE atendimento=2021;

```

5. Todos os pacientes e suas fichas médicas criadas em 2021:

$$\text{Resultado}(\text{email}, \text{idExame}, \text{idProntuário}) \leftarrow ((\text{Paciente} \bowtie (\text{email}=\text{email}) \\ \text{fichaMédica}) \cap (\text{paciente} \bowtie (\text{email}=\text{email})))$$

A equação exibe esta consulta em álgebra relacional. Em SQL temos o seguinte código:

```

1 SELECT paciente. email , idExame , idTurma
2 FROM paciente, fichaMédica, prontuário
3 WHERE paciente. email=fichaMédica. email
4 AND paciente. email=prontuário. email ;

```


6 Criação da Base de Dados

```
Create DATABASE Bancos3;

USE Bancos3;

CREATE TABLE AgenteDesaude(
cod VARCHAR(20) NOT NULL,
senha VARCHAR(5) NOT NULL,
PRIMARY KEY(cod)
);

CREATE TABLE Enfermeiro(
tip VARCHAR(20) NOT NULL,
senha VARCHAR(5) NOT NULL,
PRIMARY KEY(tip)
);

CREATE TABLE Medico(
id VARCHAR(20) NOT NULL,
senha VARCHAR(5) NOT NULL,
função INTEGER NOT NULL CHECK (função BETWEEN 1 AND 5) ,
PRIMARY KEY(id)
);

CREATE TABLE Paciente(
nome VARCHAR(30) NOT NULL,
documento VARCHAR(12) NOT NULL,
númerocartãosus VARCHAR(15) NOT NULL,
id VARCHAR(20) NOT NULL,
cod VARCHAR(20) NOT NULL,
tip VARCHAR(20) NOT NULL,
PRIMARY KEY(documento),
```

```

FOREIGN KEY(id) REFERENCES Medico(id),
FOREIGN KEY(cod) REFERENCES AgenteDesaude(cod),
FOREIGN KEY(tip) REFERENCES Enfermeiro(tip)
);

CREATE TABLE Exame(
ex VARCHAR(20) NOT NULL,
prescrição VARCHAR(30) NOT NULL,
doc VARCHAR(12) NOT NULL,
relatório VARCHAR(30) NOT NULL,
documento VARCHAR(12) NOT NULL,
PRIMARY KEY(ex),
FOREIGN KEY(documento) REFERENCES Paciente(documento)
);

CREATE TABLE Enfermidade(
en VARCHAR(30) NOT NULL,
tipo VARCHAR(30) NOT NULL,
documen VARCHAR(12) NOT NULL,
criacao DATE NOT NULL,
documento VARCHAR(12) NOT NULL,
PRIMARY KEY(en),
FOREIGN KEY(documento) REFERENCES Paciente(documento)
);

CREATE TABLE CausaAtendimento(
caus VARCHAR(30) NOT NULL,
dct VARCHAR(12) NOT NULL,
criacao DATE NOT NULL,
documento VARCHAR(12) NOT NULL,
PRIMARY KEY(caus),
FOREIGN KEY(documento) REFERENCES Paciente(documento)
);

```

```
CREATE TABLE FichaMedica(  
    fich VARCHAR(30) NOT NULL,  
    criacao DATE NOT NULL,  
    dto VARCHAR(12) NOT NULL,  
    documento VARCHAR(12) NOT NULL,  
    PRIMARY KEY(fich),  
    FOREIGN KEY(documento) REFERENCES Paciente(documento)  
);  
  
CREATE TABLE Prontuario(  
    pront VARCHAR(30) NOT NULL,  
    criacao DATE NOT NULL,  
    dpt VARCHAR(12) NOT NULL,  
    documento VARCHAR(12) NOT NULL,  
    PRIMARY KEY(pront),  
    FOREIGN KEY(documento) REFERENCES Paciente(documento) );
```