

Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Disciplina: Banco de Dados

Turma A - Professora Maristela Terto de Holanda

Projeto de Banco de Dados

Camila Imbuzeiro Camargo - 130104868

Miguel Angelo Montagner Filho - 130127302

Paulo Victor Gonçalves Farias - 130144754

Brasília, 1 de Novembro de 2015

Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Disciplina: Banco de Dados

Turma A - Professora Maristela Terto de Holanda

Sumário

Introdução

Diagrama Entidade Relacionamento

Modelo Relacional

Consultas de Álgebra Relacional

Avaliação de Formas Normais

Script SQL gerador do Banco de Dados

1. Introdução

Durante este semestre estudou-se a teoria básica que envolve a correta utilização e manipulação de um banco de dados. Com o objetivo de aplicar essa teoria é que este projeto foi desenvolvido. O grupo optou por criar um banco de dados para auxiliar na gestão do serviço público de saúde de uma determinada cidade. Os tópicos que seguem irão mostrar o processo de criação de tal banco, apresentando o Modelo Entidade Relacionamento criado como base, o Modelo Relacional derivado dele, a avaliação das formas normais de certas tabelas, o script SQL que gerou o banco de dados, e, finalmente, algumas consultas em álgebra relacional para reforçar as ideias do funcionamento de um banco de dados relacional.

É necessário levar em consideração que para este projeto o Sistema Gerenciador de Banco de Dados escolhido para implementar o banco de dados em questão foi o PostgreSQL, versão 9.4.

2. Diagrama Entidade Relacionamento

A imagem a seguir apresenta o Modelo Entidade Relacionamento criado tendo em mente o serviço público de saúde de uma cidade. Esse modelo foi criado com a ajuda da ferramenta brModelo:

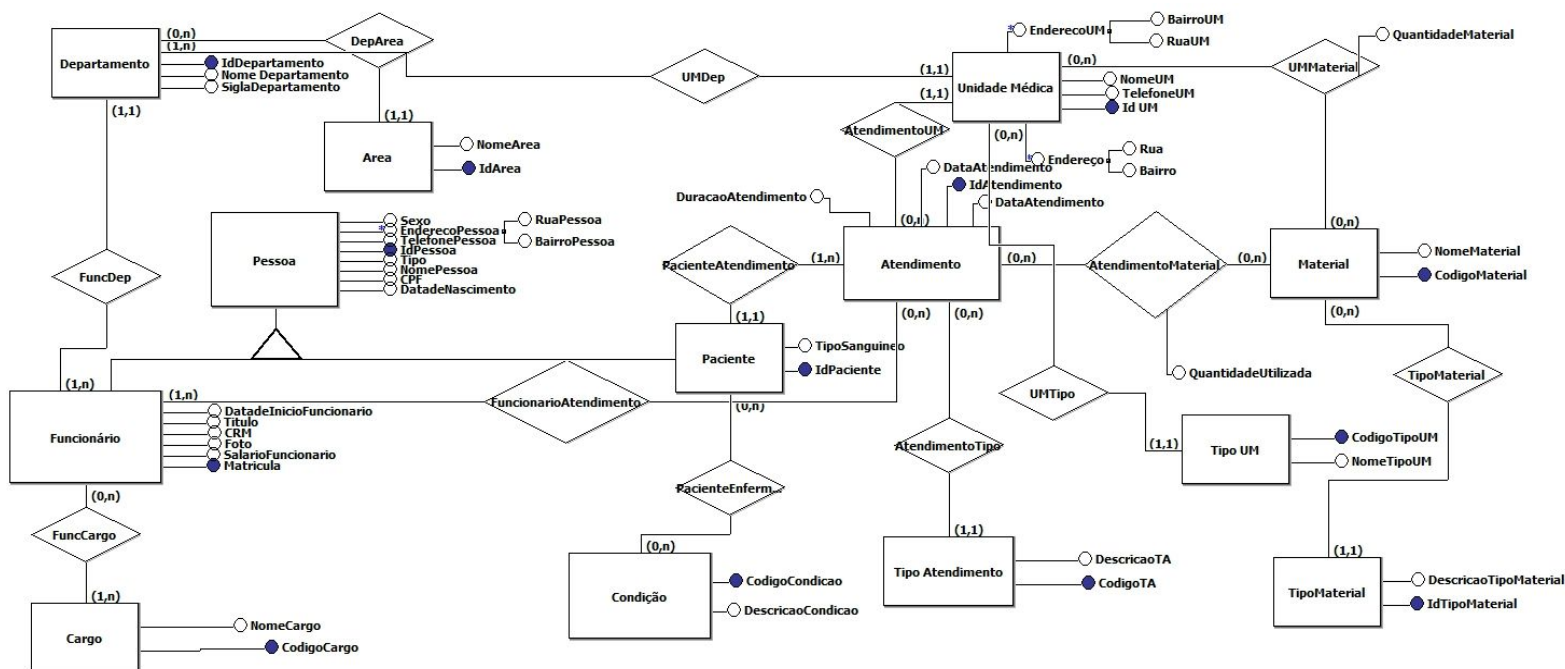


Figura 1 - Modelo Entidade Relacionamento

Esse modelo foi criado levando em consideração as seguintes premissas:

- No banco de dados teremos registros tanto de pacientes como de funcionários. Para ambos é preciso armazenar informações como o sexo, o nome, o endereço, o telefone, o CPF e a data de nascimento. Porém, para funcionários, além desses dados básicos, é necessário adicionar a matrícula do funcionário, a sua data de ingresso no serviço público, o seu salário, o seu CRM,

Para este modelo temos já identificados os tipos de cada atributo, já determinando em parte como ocorrerá a implementação deste banco de dados.

4. Consultas de Álgebra Relacional

- 1) Para os pacientes que tiveram algum tipo de atendimento, em que unidade médica ocorreu esse atendimento?

$$\pi_{\text{nomeum, nome}}(((\text{pessoa} \mid X \mid_{\text{pessoa(idpessoa) = paciente(idpessoa)}} \text{paciente}) \mid X \mid_{\text{paciente(idpaciente) = atendimento(idpaciente)}} \text{atendimento}) \mid X \mid_{\text{idum = umatendimento}} \text{unidademedica})$$

Em SQL temos:

```

SELECT DISTINCT nomeum AS "Unidade Médica", nome AS "Paciente" FROM pessoa
INNER JOIN paciente ON pessoa.idpessoa = paciente.idpessoa
INNER JOIN atendimento ON atendimento.idpaciente = paciente.idpaciente
INNER JOIN unidademedica ON unidademedica.idum = atendimento.umatendimento;

```

- 2) Quais foram os funcionários que já atenderam a paciente Migirl Angela Montagner?

$$\pi_{\text{nome}}(\sigma_{\text{atendimento(idpaciente) = 3}}(((\text{pessoa} \mid X \mid_{\text{pessoa(idpessoa) = funcionario(idpessoa)}} \text{funcionario}) \mid X \mid_{\text{funcatendimento(matr) = funcionario(matr)}} \text{fun ncatendimento}) \mid X \mid_{\text{funcatendimento(idatendimento) = atendimento(idatendimento)}}))$$

Em SQL temos:

```

SELECT DISTINCT nome AS "Nome" FROM pessoa
INNER JOIN funcionario ON pessoa.idpessoa = funcionario.idpessoa
INNER JOIN funcatendimento ON funcatendimento.matr = funcionario.matr
INNER JOIN atendimento ON atendimento.idatendimento = funcatendimento.idatendimento WHERE
atendimento.idpaciente IN (SELECT idpaciente FROM pessoa INNER JOIN paciente ON
pessoa.idpessoa = paciente.idpessoa WHERE nome LIKE 'Migirl Angela Montagner');

```

- 3) Liste o nome de todos os pacientes. Eles possuem condições?

$$\pi_{\text{pessoa(nome), condicao(descrcondicao)}}(((\text{pessoa} \mid X \mid_{\text{pessoa(idpessoa) = paciente(idpessoa)}} \text{paciente}) \mid X \mid_{\text{paciente(idpaciente) = pacientecond(idpaciente)}} \text{pacientecond}) \mid X \mid_{\text{condicao(codcondicao) = pacientecond(codcondicao)}} \text{condicao})$$

Em SQL temos:

```

SELECT pessoa.idpessoa AS "id", nome AS "Paciente", descrcondicao AS "Condição" FROM pessoa
INNER JOIN paciente ON pessoa.idpessoa = paciente.idpessoa
LEFT JOIN pacientecond ON paciente.idpaciente = pacientecond.idpaciente
LEFT JOIN condicao ON condicao.codcondicao = pacientecond.codcondicao
ORDER BY pessoa.nome;

```

- 4) Quais são as unidades médicas que possuem departamento de Urologia?

$$\pi_{sigladepartamento, nomeum, bairroum}(\sigma_{nomearea = "Urologia" ((departamento \mid_{departamento(idum) = unidademédica(idum)} unidademédica) \mid_{departamento(idarea) = area(idarea)} area))$$

Em SQL temos:

```

SELECT sigladepartamento "Departamento", nomeum AS "Unidade Médica", bairroum AS "Bairro
UM" FROM departamento INNER JOIN unidademédica
ON departamento.idum = unidademédica.idum
INNER JOIN area ON departamento.idarea = area.idarea WHERE nomearea LIKE 'Urologia' ORDER
BY unidademédica.idum;
  
```

5) Quais foram os materiais utilizados no atendimento possuindo id = 1? Qual é o código e o tipo desses materiais?

$$\pi_{codmaterial, nomematerial, descrtipomaterial}(\sigma_{idatendimento = 1}((material \mid_{material(idtipomaterial) = tipomaterial(idtipomaterial)} tipomaterial) \mid_{materialatendimento(codmaterial) = material(codmaterial)} materialatendimento)))$$

Em SQL temos:

```

SELECT material.codmaterial AS "Código", nomematerial AS "Material", descrtipomaterial AS
"Tipo Material" FROM material INNER JOIN tipomaterial
ON material.idtipomaterial = tipomaterial.idtipomaterial
INNER JOIN materialatendimento ON materialatendimento.codmaterial = material.codmaterial
WHERE idatendimento = 1;
  
```

Outras consultas foram criadas, porém, como elas referenciavam menos do que 3 tabelas elas não foram incluídas nesta parte do projeto. Para ter acesso à essas consultas verifique o arquivo “Queries.sql” presente na pasta compactada enviada em conjunto com este arquivo em PDF.

5. Avaliação de Formas Normais

Para analisar as formas normais de certas tabelas iremos primeiramente considerar como seriam representados os dados destas tabelas caso elas não estivessem dispostas em um banco de dados que surgiu de um modelo entidade-relacionamento.

Para as tabelas referentes às unidades médicas, aos departamentos e às áreas desse departamento, poderíamos ter um formulário da seguinte forma:

UNIDADE MÉDICA		
Nome:	Identificador:	Tipo:
Endereço		
Bairro:	Rua:	
Telefone:		
Departamentos		
Sigla:	Nome:	Área:
Sigla:	Nome:	Área:
Sigla:	Nome:	Área:
Sigla:	Nome:	Área:
Sigla:	Nome:	Área:

Figura 3 - Formulário hipotético

Obtendo as formas normais para este formulário teríamos a seguinte situação:

1a Forma Normal:

departamentos(iddep, nomedep, siglade, areadep, idum, nomeum, tipoum, bairroum, ruaum, telum)

2a Forma Normal:

unidade medica(idum, nomeum, tipoum, bairroum, ruaum, telum)

departamento(iddep, nomedep, siglade, areadep, idum)

idum REFERENCIA unidade medica(idum)

3a Forma Normal:

unidade medica(idum, nomeum, idend, idtipoum)

idend REFERENCIA endereco(idend)

idtipoum REFERENCIA tipoum(idtipoum)

endereco(idend, bairro, rua)

tipoum(idtipoum, nome tipoum)

departamento(iddep, nomedep, siglade, idarea, idum)

idarea REFERENCIA area(idarea)

idum REFERENCIA unidade medica(idum)

area(idarea, nome area)

Observando o modelo relacional presente na Figura 2 e fazendo uma comparação com as formas normais obtidas aqui é possível chegar à conclusão de que temos as tabelas departamento, area e tipoum já na 3a forma normal. A tabela unidade medica, apesar de normalizada, poderia ser normalizada mais ainda devido ao fato de os campos referentes ao endereço não dependerem da chave primária dessa tabela. Portanto, temos que esta tabela se encontra na 2a forma normal.

Seguindo a mesma lógica de raciocínio, podemos dizer que a tabela atendimento também está na 3a forma normal: os atributos dataatendimento, e duracaoatendimento todos dependem da chave primária de atendimento, idatendimento. A descrição do tipo de atendimento, apesar de estar relacionada com o atendimento não depende da chave primária do atendimento, afinal, podemos obter essa descrição sem que um atendimento tenha acontecido. Portanto, temos esse atributo sendo transformado em uma nova tabela. A tabela funcatendimento evita também a redundância já que um atendimento pode ter diversos funcionários envolvidos. Com essa tabela garantimos a 2a forma normal. As chaves estrangeiras referentes à tabela de paciente e unidade medica garantem que a tabela está de fato na 3a forma normal, já que atributos como o nome do paciente e sua data de nascimento, apesar de serem importantes para a tabela atendimento, não possuem dependência funcional alguma com a chave primária da tabela atendimento. O mesmo ocorre para os atributos da tabela unidade medica.

7. Script SQL gerador do Banco de Dados

Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Disciplina: Banco de Dados

Turma A - Professora Maristela Terto de Holanda

O script SQL que gerou o banco de dados pode ser encontrado no arquivo “trabalhoBD_v10.sql”, enviado juntamente com este documento escrito. Ele pode ser aberto em qualquer editor de textos para que possa ser visualizado. O script não foi diretamente colocado neste documento por ser muito grande.