Universidade Estadual de Campinas – Instituto de Computação
MC536 – Bancos de Dados – Profª. Claudia Bauzer Medeiros

Trabalho 2

agência matrimonial GetLove

Bruno Sepulveda (165207) · Giovanni Bertão (173325) · Guilherme Gama (173608)
Guilherme Perrotta (173846) · Gustavo Salibi (174135) · João Pedro Martins (176117)

ESQUEMA

ENTIDADES DO MODELO ER:

Pessoa(id_pessoa, cpf, email, nome, data_nascimento, sexo, cidade, uf, numero_filhos, tabaco, alcool, outras_drogas, religiao, tipo_de_personalidade, latitude, longitude)

Dependências funcionais:
id_pessoa → cpf, email, nome, data_nascimento, sexo, cidade, uf, numero_filhos, tabaco, alcool, outras_drogas, religiao, tipo_de_personalidade, latitude, longitude)

Dependências de inclusão:
id_pessoa deve ser não-nulo e único na coluna.
cpf, email, nome, data_nascimento, cidade, uf, sexo devem ser não-nulos.

Moderador(id_moderador, nome)

Dependência funcional:
id_moderador → nome

Dependência de inclusão:
id_moderador deve ser não-nulo e único na coluna.
nome deve ser não-nulo

Ocorrencia(id_ocorrencia, tipo, id_implicado, ref_arquivo_mensagem, data_ocorrencia)

Dependências funcionais:
id_ocorrencia → tipo, id_implicado, ref_arquivo_mensagem, data_ocorrencia

Dependências de inclusão:
id_ocorrencia deve ser não-nulo e único na coluna.
id_implicado, ref_arquivo_mensagem, data_ocorrencia devem ser não-nulos

Empresa(id_empresa, cnpj, nome, email, telefone, cidade, uf)

Dependências funcionais:
id_empresa → cnpj, nome, email, telefone, cidade, uf

Dependências de inclusão:
id_empresa deve ser não-nulo e único na coluna.

Empresa_de_dados(id_empresa)

Dependências funcionais:
não há

Dependências de inclusão:
id_empresa deve ser chave de alguma tupla em **Empresa**.

Empresa_de_servicos(id_empresa)

Dependências funcionais:
não há

Dependências de inclusão:
id_empresa deve ser chave de alguma tupla em **Empresa**.

RELACIONAMENTOS NO MODELO ER:

Combina_com(id_pessoa1, id_pessoa2, ref_arquivo_conversa, data_combinacao)

Dependências funcionais:
id_pessoa1, id_pessoa2 → ref_arquivo_conversa, data_combinacao

Dependências de inclusão:
A chave composta id_pessoa1, id_pessoa2 deve ser não-nula e única na coluna.
ref_arquivo_conversa e data_combinacao devem ser não-nulos

Aprova(id_pessoa, id_aprovado, data_aprovacao)

Dependências funcionais:
id_pessoa1, id_pessoa2 → data_aprovacao

Dependências de inclusão:
A chave composta id_pessoa1, id_pessoa2 deve ser não-nula e única na coluna.
data_aprovacao deve ser não-nula

Tem_dados_vendidos_a(id_pessoa, id_empresa)

Dependências de inclusão:
A chave composta id_pessoa1, id_empresa deve ser não-nula e única na coluna.

Abre(id_ocorrencia, id_pessoa)

Dependências funcionais:
id_ocorrencia → id_pessoa

Dependência de inclusão:
id_ocorrencia deve ser não-nulo e único na coluna

Avalia(id_ocorrencia, id_moderador)

Dependências funcionais:
id_ocorrencia → id_moderador

Dependência de inclusão:
id_ocorrencia deve ser não-nulo e único na coluna

ENTIDADES FRACAS NO MODELO ER:

Plano_premium(id_pessoa, data_expiracao)

Dependências funcionais:
id_pessoa → data_expiracao

Dependências de inclusão:
id_pessoa deve ser chave de tupla na tabela Pessoa

Servico(id_empresa, numero_servico, preco, categoria, ref_arquivo_descricao, latitude, longitude)

Dependências funcionais:
id_empresa, numero_servico → preco, categoria, ref_arquivo_descricao, latitude, longitude

Dependências de inclusão:
id_empresa deve ser chave de tupla na tabela Empresa_de_Servicos
a chave composta (id_empresa, numero_servico) deve ser não-nula e única na coluna

Foto(id_pessoa, numero_foto, ref_arquivo_imagem)

Dependências funcionais:
id_pessoa, numero_foto → ref_arquivo_imagem

Dependências de inclusão:
id_pessoa deve ser chave de tupla na tabela Pessoa
a chave composta (id_pessoa, numero_servico) deve ser não-nula e única na coluna

ref_arquivo_imagem deve ser única na coluna e não-nula

Imagem_servico(id_empresa, numero_servico, numero_imagem, ref_arquivo_imagem)

Dependências funcionais:
id_empresa, numero_servico, numero_imagem → ref_arquivo_imagem

Dependências de inclusão:
a chave parcial (id_empresa, numero_servico) deve ser chave de tupla na tabela Servico
a chave composta (id_empresa, numero_servico, numero_imagem) deve ser não-nula e única na coluna

RELACIONAMENTOS FRACOS NO MODELO ER:
implementadas em SQL por chave estrangeira para o dono (após minimização).

É assinado(id_pessoa)
É imagem de(id_produto, numero_imagem)
Tem_foto(id_pessoa, numero_foto)
Oferece(id_empresa, numero_servico)

Lista de operações

(especificação completa das operações foi refeita no T1 corrigido.)

Consultas:
Encontrar pessoas
Gerar encontro
Listar combinações
Listar quem me aprovou
Recuperar log de conversa

Listar ocorrências abertas por cliente x:

$$\sigma_{id\_pessoa = id\_cliente\_x}(Abre)$$

Listar ocorrências sobre cliente x:

$$\sigma_{tipo = 0 \wedge id\_implicado = id\_cliente\_x}(Ocorrencia)$$

Listar ocorrências sobre a empresa x:

$$\sigma_{tipo = 1 \wedge id\_implicado = id\_empresa\_x}(Ocorrencia)$$

Listar ocorrências sobre a getLove:

$$\sigma_{tipo = NULL}(Ocorrencia)$$

Listar ocorrências abertas:

$$Ocorrencia \bowtie ((\pi_{id\_ocorrencia}(Ocorrencia)) - (\pi_{id\_ocorrencia}(Avalia)))$$

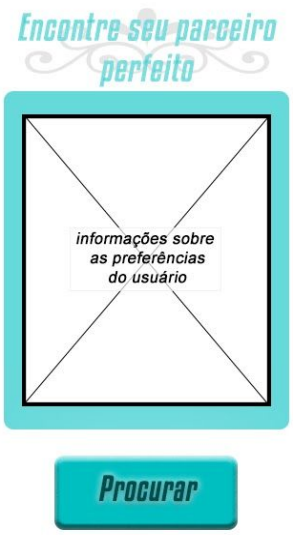
Atualizações:
Remover combinação
Alterar perfil
Refazer teste de personalidade
Recalcular posição geográfica
Alterar fotos
Alterar plano
Alterar lista de empresas parceiras
Alterar lista de produtos
Alterar imagem de produto

Questões de implementação

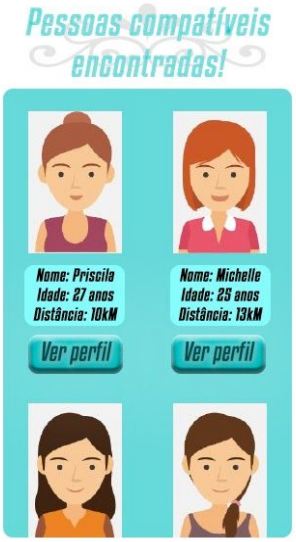
1. Optamos por demonstrar o uso do banco de dados utilizando comandos SQL.

A demonstração se dará através de uma plataforma fictícia, onde o sistema executaria funcionalidades específicas além dos comandos SQL, tais como: fazer combinação entre pessoas, exibir ao usuário quem demonstrou interesse por ele, selecionar melhor opção de encontro a ser recomendado pela plataforma, funcionalidade de escrever e enviar mensagens, uma tela para denunciar outros usuários, além de possibilidade de atualizar perfil alterando informações. Seguem abaixo algumas imagens que ilustram como seria tal plataforma:

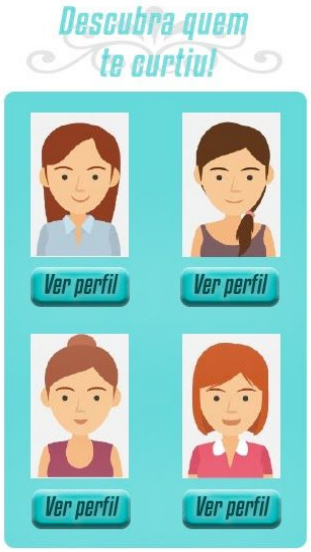
1.



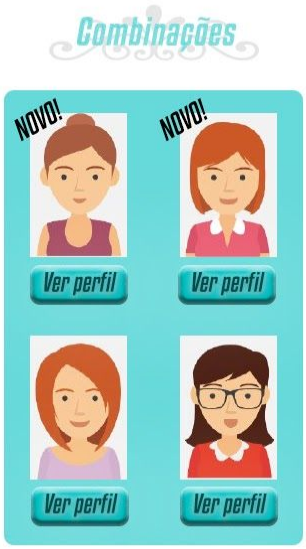
2.



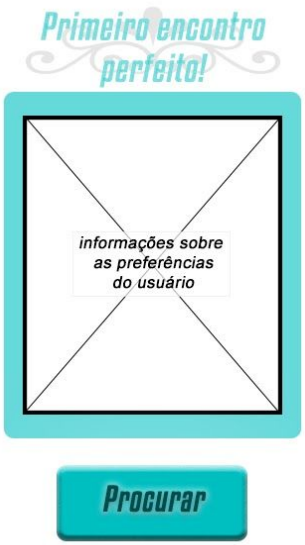
3.



4.



5.



6.

2. O banco de dados será criado utilizando os seguintes comandos do SGBD MySQL:

```
create database GetLove;
use GetLove;

create table Pessoa (
    id_pessoa int primary key,
    cpf char(11) not null unique,
    email varchar(30) not null unique,
    nome varchar(100) not null,
    data_nascimento date not null,
    sexo char not null,
    cidade varchar(30) not null,
    uf char(2) not null,
    numero_filhos int,
    tabaco int,
    alcool int,
    outras_drogas int,
    religiao int,
    tipo_de_personalidade int,
    latitude float(10,6),
    longitude float(10,6)
);

create table Moderador (
    id_moderador int primary key,
    nome varchar(100) not null
);

create table Ocorrencia (
    id_ocorrencia int primary key,
    tipo char(7) not null,
    id_implicado int,
    ref_arquivo_mensagem varchar(50) not null,
    data_ocorrencia date not null,
    check (tipo like "getlove" or tipo like "usuario" or tipo like "empresa")
);

create table Empresa (
    id_empresa int primary key,
    cnpj char(14) not null unique,
    nome varchar(100) not null,
    email varchar(30) not null unique,
    telefone varchar(11) not null unique,
    cidade varchar(30) not null,
    uf char(2) not null
);
```

```sql
create table Empresa_de_dados (
    id_empresa int not null,
    foreign key(id_empresa) references Empresa(id_empresa)
);

create table Empresa_de_servicos (
    id_empresa int not null,
    foreign key(id_empresa) references Empresa(id_empresa)
);

create table Combina_com (
    id_pessoa1 int,
    id_pessoa2 int,
    ref_arquivo_conversa varchar(50) not null,
    data_combinacao date not null,
    primary key(id_pessoa1, id_pessoa2)
);

create table Aprova (
    id_pessoa int,
    id_aprovado int,
    data_aprovacao date not null,
    primary key(id_pessoa, id_aprovado)
);

create table Tem_dados_vendidos_a (
    id_pessoa int,
    id_empresa int,
    primary key(id_pessoa, id_empresa)
);

create table Abre (
    id_ocorrencia int,
    id_pessoa int,
    primary key(id_ocorrencia)
);

create table Avalia (
    id_ocorrencia int,
    id_moderador int,
    primary key(id_ocorrencia)
);

create table Plano_premium (
    id_pessoa int not null,
    data_expiracao date not null,
    foreign key(id_pessoa) references Pessoa(id_pessoa)
);
```

```
create table Servico (
    id_empresa int not null,
    numero_servico int not null,
    preco float(10,2),
    categoria varchar(5),
    ref_arquivo_descricao varchar(50),
    latitude float(10,6),
    longitude float(10,6),
    foreign key(id_empresa) references Empresa_de_servicos(id_empresa),
    primary key(id_empresa, numero_servico)
);

create table Foto (
    id_pessoa int not null,
    numero_foto int not null,
    ref_arquivo_imagem varchar(50) not null unique,
    foreign key(id_pessoa) references Pessoa(id_pessoa),
    primary key(id_pessoa, numero_foto)
);

create table Imagem_servico (
    id_empresa int not null,
    numero_servico int not null,
    numero_imagem int not null,
    ref_arquivo_imagem varchar(50) not null,
    foreign key(id_empresa, numero_servico) references Servico(id_empresa,
numero_servico),
    primary key(id_empresa, numero_servico, numero_imagem)
);
```

3.

a) O banco de dados será alimentado pela importação de arquivos csv. O conteúdo desses arquivos virá de python scripts criados por nós. Esses scripts devem gerar grande quantidade de entidades aleatórias com atributos advindos de outros bancos de dados (como o atributo nome, de Pessoa) e alguns gerados aleatoriamente (como os atributos numéricos) para criar os arquivos .csv.

b) A tabela 1 lista o número de tuplas esperadas para cada relação .

c) Dificuldades imaginadas na implementação:

- Calcular distância entre clientes ou entre clientes e produtos, no intuito de alocar ou sugerir serviços que estejam próximos aos usuários envolvidos.
- Encontrar bancos de dados grandes para download que forneçam os dados necessários para geração das entidades que serão utilizadas para alimentar o banco de dados da aplicação, mencionadas em (a).
- Criar os python scripts para gerar os arquivos .csv mencionados em (a), com várias tuplas de todas as entidades, que possuem muitos atributos cada e com bastante variação de tipos.

Tabela 1 – Listagem da quantidade prevista de tuplas para cada relação criada

Relação	Quantidade prevista (lim. superior)
Pessoa	500
Ocorrencia	50
Moderador	10
Empresa	50
Plano_premium	200
Servico	100
Foto	1000
Imagem_servico	200
Combina_com	1250
Aprova	2500
Tem_dados_vendidos_a	5000*
Eh_assinado	200
Eh_imagem_de	200
Tem_foto	1000
Oferece	100
Abre	50
Avalia	25

*: Diversos fatores podem influenciar esse valor, uma vez que qualquer tipo de dado pode ser vendido a qualquer empresa interessada. Contudo, pela natureza do programa consideramos os seguintes dados: Produtos recém adquiridos e produtos visualizados. Tais dados, se baseiam em uma aproximação inferior diferente das outras linhas que estão relacionadas ao limite superior.

Mudanças de rumo

Serviço Casamento dos Sonhos foi removido para focarmos nos serviços essenciais, então as queries “adicionar noivado” e “gerar cerimônia de casamento” também foram removidas.

Primeiro Encontro Perfeito e o serviço de encontrar parceiros em potencial agora levam a distância em consideração, necessitando que a aplicação grave dados de coordenada geográfica no banco de dados.