



Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

Primeiro Trabalho Prático
Prof. Dr. Caetano Traina Júnior

Esse trabalho foi desenvolvido pelos alunos:

- João Pedro Soares de Azevedo Calixto | nroUSP: 13732011 | email: jpcalixto@usp.br
- Miller Matheus Lima Anacleto Rocha | nroUSP: 13727954 | email: AnacletoMiller@usp.br

Descrições e restrições das tabelas

Usuário

Descrição

Cabe a essa entidade o armazenamento de informações pessoais de qualquer usuário da plataforma, sendo esse locatário ou anfitrião. Com essa entidade conseguimos reduzir duplicações de informação no caso de usuários que são anfitriões e locatários ao mesmo tempo. Além disso, o uso dessa tabela também possibilitou maior atomicidade e organização do projeto como um todo.

Restrições

O id é gerado automaticamente e é a chave primária que identifica unicamente cada usuário. Nome de usuário, sobrenome, número de telefone, endereço (formado por cidade, estado, cep, rua, número da cada e complemento) são campos obrigatórios para essa tabela, enquanto o resto dos campos são opcionais.

Mensagem

Essa tabela armazena mensagens trocadas entre usuários (sendo eles anfitriões ou locatários). É obrigatório identificar o usuário que enviou, o usuário que recebeu, o conteúdo e o timestamp da mensagem.

Locatário

Descrição

Cabe a essa entidade identificar unicamente um usuário que é um locatário. Se há um usuário que é locatário, então essa tabela identifica-o unicamente pelo id do usuário (user_id).

Restrição

Como todo locatário é necessariamente um anfitrião, então não há restrições específicas para essa tabela. Por conta da necessidade de um usuário, essa será uma entidade fraca a qual depende da existência de um usuário.



Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

Primeiro Trabalho Prático
Prof. Dr. Caetano Traina Júnior

Esse trabalho foi desenvolvido pelos alunos:

- João Pedro Soares de Azevedo Calixto | nroUSP: 13732011 | email: jpcalixto@usp.br
- Miller Matheus Lima Anacleto Rocha | nroUSP: 13727954 | email: AnacletoMiller@usp.br

Anfitrião

Descrição

Cabe a essa entidade identificar unicamente um usuário que é um anfitrião. Se há um usuário que é anfitrião, então essa tabela identifica-o unicamente pelo id do usuário (user_id).

Restrições

Assim como no caso do locatário, não há restrições específicas para essa tabela. Pelo mesmo motivo do locatário, essa será uma entidade fraca.

Conta

Descrição

Essa tabela identifica as informações bancárias de um anfitrião

Restrições

É necessário preencher todos os campos dessa tabela (nenhum campo pode ser nulo), além de que essa tabela faz referência unicamente a um anfitrião. Por conta da necessidade de um anfitrião, essa será uma entidade fraca que sua existência depende da existência de um anfitrião.

Propriedade

Descrição

Essa tabela armazena informações sobre as propriedades disponibilizadas pelos anfitriões.

Restrições

O campo id é gerado automaticamente. O único campo com preenchimento opcional é o complemento, os demais campos são todos obrigatórios.



Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

Primeiro Trabalho Prático
Prof. Dr. Caetano Traina Júnior

Esse trabalho foi desenvolvido pelos alunos:

- João Pedro Soares de Azevedo Calixto | nroUSP: 13732011 | email: jpcalixto@usp.br
- Miller Matheus Lima Anacleto Rocha | nroUSP: 13727954 | email: AnacletoMiller@usp.br

Quarto, Comodidade e Regra

Essas tabelas são tabelas relacionadas à tabela Propriedade. Sendo assim, possuem informações pertinentes a cada propriedade. Cada uma dessas tabelas deve ter em cada uma de suas tuplas, uma referência a qual propriedade pertencem unicamente, sendo que uma propriedade pode ter referência a várias comodidades, regras e quartos específicos (a volta não vale - isto é, um quarto, comodidade ou regra não podem fazer referência a mais de uma propriedade). Por conta da necessidade de uma propriedade, comodidade e regra são entidades fracas. Quarto não foi selecionado como entidade fraca, pois pretende-se manter essa tabela mesmo que não haja a propriedade a qual o quarto faz referência.

Cama

Essa tabela faz referência a tabela Quarto, sendo que um quarto pode possuir uma ou mais camas. Colocamos essa tabela separada, pois o *tipo de cama* na descrição do projeto pareceu genérico (podendo ser cama de solteiro, cama de casal, cama king, cama redonda, etc...). Por conta desse tipo genérico, foi mais sensato deixar uma tabela própria para a cama. Por conta da necessidade de um quarto, a cama é uma entidade fraca.

Avaliação

Descrição

Essa tabela armazena avaliações feitas pelos locatários sobre as propriedades. Um locatário pode fazer zero ou várias avaliações as quais fazem referência cada uma a uma única propriedade. A propriedade, no entanto, possui cada uma várias avaliações feitas cada uma por um único locatário. Além disso, cada avaliação possui cada uma zero ou várias fotos referentes.

Restrições

É obrigatório conter em cada avaliação um identificador sobre qual propriedade se faz referência e um identificador sobre qual locatário fez aquela avaliação. Além disso, é necessário também preencher as notas referentes a essa avaliação.

Foto

Em cada tupla da tabela, há uma foto que faz referência unicamente a uma avaliação. Por conta da necessidade de uma avaliação, essa será uma entidade fraca.



Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

Primeiro Trabalho Prático
Prof. Dr. Caetano Traina Júnior

Esse trabalho foi desenvolvido pelos alunos:

- João Pedro Soares de Azevedo Calixto | nroUSP: 13732011 | email: jpcalixto@usp.br
- Miller Matheus Lima Anacleto Rocha | nroUSP: 13727954 | email: AnacletoMiller@usp.br

Reserva

Descrição

Essa tabela armazena informações sobre as reservas feitas pelos locatários a determinadas propriedades. Cada propriedade possui uma janela de disponibilidade descrita pelos campos `data_check_in`, `data_check_out` e `status_reserva`.

Supondo que um anfitrião coloque sua propriedade a uma janela de segunda à sexta como disponível. Então, essa propriedade terá a data de check in e check out configurada para iniciar na segunda e terminar na sexta e com `status_reserva` `False` - isto significa que a propriedade se encontra disponível para reservas de segunda a sexta. Se algum locatário escolher reservar de segunda a quarta, então essa tupla que referenciava essa disponibilidade irá se dividir em duas: a primeira com data check in e check out começando na segunda e indo até quarta, com `status_reserva` = `True` e fazendo referência ao locatário que fez essa reserva; já a outra dupla terá data check in e check out começando na quarta depois do horário de check out e terminando na sexta com `status_reserva` = `False` (indicando a nova disponibilidade dessa propriedade nesses dias).

Restrições

É necessário identificar a propriedade a qual se faz referência bem como as datas e horários de check in e check out e preço relativo a essa janela de disponibilidade.