



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

Primeiro Trabalho Prático  
Prof. Dr. Caetano Traina Júnior  
Entrega dia 16 de maio de 2024

## 1 Descrição do problema

Uma empresa provê um serviço online para que pessoas anunciem, pesquisem e reservem propriedades para acomodação e meios de hospedagem em localidades de todo o mundo. Ela permite que indivíduos possam alugar no todo ou em parte seus imóveis, incluindo sua própria casa, como uma forma de rendimento extra. A empresa fornece uma plataforma de busca e reservas entre a pessoa que oferece a acomodação e o viajante que procura um local para acomodação em locação. Este trabalho visa criar uma base de dados para armazenar as informações envolvidas nas atividades da empresa.

Dentre essas informações está a descrição das **propriedades**, que deve estar identificada por pelo menos com um **nome e endereço**. Cada propriedade é classificada como sendo uma **casa inteira** ou **um quarto**, que pode ser **individual ou compartilhado**, e sua descrição deve incluir o **número de quartos e de banheiros**, preço por noite, número máximo de hóspedes permitidos e números mínimos e máximos de noites que podem ser alugadas, e pode indicar ainda o valor da taxa de limpeza, datas em que a propriedade está disponível, horário de **check-in** e **check-out**. Cada **quarto de uma propriedade** deve indicar **quantas camas ele tem e o tipo de cama em cada quarto**. Cada **propriedade pode ter diversas regras** (se é permitido fumar, animais de estimação, etc.) e diversas comodidades (como WiFi, ar-condicionado etc.)

A empresa contempla dois tipos de **usuários**: o **anfitrião**, que é quem loca a propriedade, e o **locatário** (hóspede). Todos têm um **nome, sobrenome, data de nascimento, endereço, sexo, número de telefone, endereço de e-mail e a senha**. Assuma que não existem dois usuários com o mesmo nome, sobrenome e número de telefone.

Cada locação é feita por meio de uma **reserva de estadia**, sendo esta feita por um locatário **para uma propriedade em uma data de check-in**. Cada locação deve incluir também uma **data da reserva** (quando a reserva foi feita), a data de **check-out** e o número de hóspedes na locação, e pode incluir o imposto pago, o preço total da estadia, o preço total com impostos (e taxa de limpeza), bem como um código promocional e o valor do desconto (se alguma promoção foi aplicada). Note-se que cada locação se refere somente a uma propriedade e um locatário. Uma reserva pode ser confirmada ou cancelada até uma data anterior à data do check-in, e se confirmada, deve ter seu pagamento efetuado pelo locatário. Cada propriedade pode ter muitas reservas, mas o período da reserva (entre as datas de check-in e check-out) não pode se sobrepor aos períodos das demais reservas dessa propriedade.

Cada propriedade é de apenas um **anfitrião**, e cada anfitrião tem uma **conta bancária onde será depositado o valor da estadia do locatário em cada reserva, contendo um número de conta exclusivo, um número de roteamento e o tipo da conta**. Uma **conta bancária só pode pertencer a um anfitrião**. Além disso, **cada propriedade e cada usuário estão associados a uma localização, que corresponde à cidade, estado e país correspondente, e possivelmente a um bairro e/ou um ou mais pontos de interesse, os quais também devem estar associados a uma única localização.**

A avaliação do sistema é feita pelas mensagens enviadas pelos usuários. Um anfitrião pode enviar 0 ou mais mensagens para um locatário e vice-versa. Cada mensagem é identificada por quem a enviou e o timestamp de sua criação, e deve incluir para quem foi enviada e o texto da mensagem. Um locatário pode avaliar zero ou mais propriedades e a propriedade pode conter zero ou mais avaliações dos locatários. Cada avaliação pode incluir uma mensagem, zero ou mais fotos e uma classificação de limpeza e uma nota para a comunicação com o anfitrião, para a localização e para o valor da locação.

## 2 Tarefas

### 2.1 Modelagem E-R

---

**Exercício 1:** Faça a modelagem E-R da base de dados descrita, usando apenas os conceitos do Modelo Entidade-Relacionamento Básico.

Nota: Para isso, você pode usar um editor gráfico genérico ou um editor específico para DE-Rs tais como Draw io (<https://app.diagrams.net>), ERDPlus (<https://erdplus.com>), etc.

### 2.2 Mapeamento ME-R→MRel

---


**Exercício 2:** Faça o Mapeamento ME-R→MRel da modelagem obtida no Exercício 1, incluindo as restrições de integridade pertinentes.

### 2.3 Consultas em SQL

---

**Exercício 3:** Crie um arquivo de tipo texto como um *script* “.sql” contendo os comandos de criação das tabelas definidas no Exercício 2.

Não é necessário mapear as restrições de integridade.

A base de dados pode ser criada tanto na base *default* do gerenciador que você estiver usando (em princípio ) quanto em uma base criada especialmente para ela (como mostrado em aula).

Use como referência o *script* estudado na célula executável “4. Criar tabelas em uma Base de Dados” do notebook Jupyter “01-PreparaBase Alunos15.ipynb” mostrado em sala de aula.

---

**Exercício 4:** Faça uma carga com dados fictícios para alimentar a base de dados criada no exercício anterior.

Crie um arquivo de tipo texto como um *script* “.sql” contendo os comandos de carga das tabelas, usando como referência o *script* estudado na célula executável “5. Carregar os dados nas tabelas” do notebook Jupyter “01-PreparaBase Alunos15.ipynb” mostrado em sala de aula.

- As relações que mapeiam Conjuntos de Entidades devem conter pelo menos 10 tuplas;
- As relações que mapeiam Conjuntos de Relacionamentos devem conter uma quantidade de tuplas que permita ilustrar adequadamente os comandos de consulta que serão emitidos sobre elas.
- Contemplar alguns atributos nulos naqueles que podem ser deixados nulos.

---

**Exercício 5:** Mostre o conteúdo feito para a relação que implementa o conceito de **Propriedades** do sistema:

- Mostre a relação inteira;
- Mostre quantas Propriedades existem de cada classe (casa inteira, etc.);
- Mostre quantas Propriedades existem de cada Cidade.

---

**Exercício 6:** Mostre as locações que foram confirmadas com *check-in* a partir de 2024-04-01, indicando para cada uma:

- Todos os atributos-chave das relações envolvidas,
- O total de dias locado,
- O Nome do anfitrião e o do locatário,

- O valor da diária.

---

**Exercício 7:** Mostre o valor médio das diárias de todas as locações que foram feitas e das que foram confirmadas em cada mês para o qual exista alguma locação na base.

---

**Exercício 8:** Mostre os anfitriões que tiveram pelo menos 3 locações, mostrando seu nome, sua cidade e quantidade de imóveis dos quais ele é dono.

---

**Exercício 9:** Mostre os usuários que são tanto anfitriões quanto locatários.

---

**Exercício 10:** Mostre:

- Os locatários que são mais jovens do que algum anfitrião.
- Os locatários que são mais jovens do que todos os anfitriões.

### 3 Regras gerais

Este trabalho deve ser feito em grupo de quatro alunos, e entregue dia 16 de maio de 2024 (quinta-feira). Identifique todos os participantes pelo seu nome, número USP e e-mail.

A entrega do trabalho deve ser feita pelo Tidia, em um arquivo em formato .zip contendo: para cada tarefa:


**Tarefa 1:** Modelagem E-R

1. Um arquivo em formato .pdf, com o Diagrama Entidade-Relacionamento do projeto conceitual do problema;  
Sugestão: use ferramentas de diagramação tal como
2. Um arquivo em formato .pdf, com a descrição das decisões de projeto/restrições assumidas pela equipe para cada Conjunto de entidades e de relacionamentos, se necessário.

**Tarefa 2:** Mapeamento E-R → MRel

1. Um arquivo em formato .pdf, com o mapeamento do diagrama da Tarefa 1 para o Modelo Relacional. Esse arquivo deve ter uma página para cada Conjunto de entidades e de relacionamentos com as respectivas anotações de mapeamento e informações de implementação, se necessário.

**Tarefa 3:** Consultas 3 a 10

1. Um arquivo em formato texto .sql com os comandos em SQL;
2. Um arquivo em formato .pdf, contendo
  - Um comentário que explique como cada comando funciona (documentação do comando); e
  - Uma listagem do resultado da execução do(s) comando(s) em  PostgreSQL. Quando o resultado tiver até 20 tuplas, listar todas elas. Quando o resultado tiver mais de 20 tuplas, listar apenas entre 10 a 20 das tuplas mais significativas, que mostrem tuplas de maior interesse para mostrar a corretude do comando executado.

Portanto, o arquivo `.zip` deve conter 19 arquivos, sendo onze `.pdf` e oito `.sql`. Alguns exercícios podem consistir de mais de um comando em SQL. Nesse caso, coloque todos eles no mesmo arquivo (`.pdf` e `.sql`)

A nota será atribuída valendo 2 para a tarefa 1, 1 para a tarefa 2, e 7 para a tarefa 3 com o mesmo peso para todos os exercícios.

[Última atualização desta lista: 1 de maio de 2024]