



**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 o Modelo Entidade-Relacionamento Estendido (ME-RX), identificando pelo menos uma Abstração de Generalização e uma de Agregação.

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:

- Todas as restrições de integridade pertinentes;
- Identifique os atributos que possam ser considerados compostos ou multi-valorados, e efetue o mapeamento correspondente;
- Identifique possíveis alternativas de mapeamento, descreva cada uma, e escolha a mais indicada para mapear neste caso.
- Escreva os comandos em SQL que criam as tabelas obtidas.

### 2.3 Carga dos arquivos a partir de um *dataset* obtido na WEB e Normalização dos dados

Esta tarefa é feita a partir de um conjunto de dados real disponibilizado na WEB pela empresa  **airbnb**, acessíveis pelos links <https://insideairbnb.com/get-the-data/> e <https://insideairbnb.com/rio-de-janeiro>, para um período de três meses de atividades do *site*.



---

**Exercício 3:** Obtenha os arquivos [Calendar.csv](#), [Listings.csv](#) e [Reviews.csv](#), prepare 3 tabelas para receber os dados dos arquivos, e faça a carga de cada tabela a partir dos arquivos respectivos.

---

**Exercício 4:** Normalize as 3 tabelas para BCNF, usando como referência para as dependências funcionais, aquelas identificadas no DE-RX e no mapeamento efetuados nos exercícios anteriores.

---

**Exercício 5:** Correlacione todos os dados existentes nas tabelas geradas pelo mapeamento e naquelas geradas pela carga de dados do  **airbnb**, indicando quais estão presente em ambos os conjuntos de tabelas, e quais estão apenas nas tabelas do mapeamento e quais apenas da carga. Faça a carga de cada tabela gerada pelo mapeamento a partir dos processos de mapeamento das tabelas obtidas pela carga do  **airbnb**.

## 2.4 Consultas em SQL

Todas as consultas a seguir devem ser executadas usando apenas os dados disponíveis nas tabelas geradas pelo Mapeamento no exercício 2, com os dados carregados no exercício 3.

---

**Exercício 6:** Mostre o conteúdo feito para a relação que implementa o conceito de **Propriedades** do sistema:

- Mostre a estrutura da relação, mostrando os atributos de 10 tuplas aleatoriamente;
- Mostre quantas Propriedades existem de cada classe (casa inteira, etc.);
- Mostre quantas localizações existem na base (usando os dados obtidos a partir do atributo `Listings.neighbourhood_cleansed`).

---

**Exercício 7:** Crie uma visão chamada `LocacaoEfetuada` que Mostra 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.

- Crie o comando da visão.
- Execute uma consulta usando essa visão que mostre as locações efetuadas em fevereiro.

---

**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.
- Crie um índice que agilize a execução dessa consulta.

## 3 Regras gerais

Este trabalho deve ser feito em grupo de quatro alunos, e entregue dia 28 de junho 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;
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 as alternativas de mapeamento que foram avaliadas e as que foram efetivamente aplicadas.
2. Um arquivo em formato `.sql`, com os comandos de criação das tabelas mapeadas.

**Tarefa 3:** Carga e normalização da base de dados a partir dos dados do  **airbnb**.

1. Um arquivo em formato **.pdf**, com uma tabela correlacionando os atributos existentes nas tabelas geradas pelo mapeamento e pela carga de dados.
2. Um arquivo em formato **.pdf**, indicando os processos de normalização executados sobre as tabelas do  **airbnb**.
3. Um arquivo em formato **.sql**, com os comandos de criação e carga das tabelas do  **airbnb**.
4. Um arquivo em formato **.sql**, com os comandos de carga das tabelas mapeadas.

**Tarefa 3:** Consultas 4 a 6

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 10 tuplas, listar todas elas. Quando o resultado tiver mais de 10 tuplas, listar apenas entre 5 a 10 das tuplas mais significativas, que mostrem tuplas de maior interesse para mostrar a corretude do comando executado.

Portanto, o arquivo **.zip** deve conter 14 arquivos, sendo sete **.pdf** e sete **.sql**.

A nota será atribuída valendo **dois** para as tarefas 1 e 2 e **três** para as tarefas 3 e 4, com o mesmo peso para todos os exercícios dentro de cada tarefa.

[Ultima atualização desta lista: 10 de junho de 2024]