

Para esse mapeamento, todas as entidades do diagrama entidade-relacionamento foram transformadas em tabela, e todas essas por sua vez utilizam de uma chave artificial para chave primária.

## Users

Todas as contas de usuário (anfitriões e hosts) foram unificadas na tabela Users, explicitando o tipo de conta no atributo User\_TYPE, conforme:

Users(User\_id, Phone\_number, Email, User\_name, User\_TYPE, User\_password)

Essa escolha foi embasada no fato de anfitriões e hosts não possuírem atributos além dos herdados por serem usuários. Além disso, foi utilizada uma chave artificial User\_id pois uma pessoa pode criar várias contas, e dessa forma a identificação é facilitada.

## Countries/States/Cities/Location

O mapeamento foi realizado tendo em vista que:

- Uma cidade (Cities) está contida em somente um estado (States);
- Um estado (States) está contido em somente um país (Countries);
- O anfitrião pode cadastrar arbitrariamente endereços (Locations), mas deve selecionar cidades, estados e países previamente listados nas tabelas Countries, States e Cities.

Com isso em mente, optou-se por criar quatro tabelas distintas, conforme:

1. Countries(Country\_id, Country\_name)
2. States(State\_id, State\_name, Country\_id\*)  
Country\_id referencia Country\_id(Countries)
3. Cities(City\_id, City\_name, State\_id\*)  
State\_id referencia State\_id(States)
4. Locations (Location\_id, Address, Location\_number, City\_id\*, State\_id\*, Country\_id\*)  
City\_id referencia City\_id(Cities)  
State\_id referencia State\_id(States)  
Country\_id referencia Country\_id(Countries)

Todas as tabelas possuem chaves artificiais, a fim de otimizar a pesquisa utilizando números (em vez de char), já que nenhuma das entidades possuía um atributo único em formato numérico.

## Accommodation

As acomodações ofertadas podem ser múltiplas por anfitrião, então para a identificação dessas, uma chave artificial se faz necessária, e por isso foi criado o atributo Accommodation\_id.

Accommodations(Accommodation\_id, Price, Has\_garage, Number\_of\_bedrooms, Number\_of\_bathrooms, Has\_wifi, Description, User\_id\*, Location\_id\*)  
User\_id referencia User\_id(Users)  
Location\_id referencia Location\_id(Locations)

## Reservations

As reservas podem ser múltiplas por anfitrião, hóspede e acomodação, então para a identificação dessas, uma chave artificial se faz necessária, e por isso foi criado o atributo Reservation\_id.

Reservations(Reservation\_id, Check-in\_date, Number\_of\_guests, Checkout\_date, Final\_price, Was\_approved, Accommodation\_id\*, Guest\_id\*)  
Accommodation\_id referencia Accommodation\_id(Accommodations)  
Guest\_id referencia User\_id(Users)

## Favorites

Para modelar o relacionamento (n-m) entre acomodações e listas de favoritos, optou-se por criar uma tabela com chave artificial sequencial e numérica, já que nenhum dos atributos, neste caso, é único e a combinação de atributos como chave primária diminuiria o desempenho das consultas.

Favorites(Favorites\_id, Accommodation\_id\*, User\_id\*, Favorites\_name)  
Accommodation\_id referencia Accommodation\_id(Accommodations)  
User\_id referencia User\_id(Users)

## Inclusion/Messages

Ambos os relacionamentos “Inclusion” e “Messages”, por se tratarem de relacionamentos n-m, precisaram virar tabela para comportar os dados envolvidos.  
Para inclusion, não utilizamos uma chave artificial como chave primária, e sim a união dos atributos Favorites\_id e Accommodation\_id. Já em messages, uma chave artificial foi criada.

Inclusion(Favorites\_id\*, Accommodation\_id\*)  
Favorites\_id referencia Favorites\_id(Favorites)  
Accommodation\_id referencia Accommodation\_id(Accommodations)

Messages(Messages\_id, Message\_timestamp, Message\_content, User\_id\*, User\_id\*)