

Normalização

Artur De Vlieger Lima [13671574], Gabriel da Costa Merlin [12544420],
Gustavo Gabriel Ribeiro [13672683], Lucas de Souza Brandão [13695021]

1. Regras funcionais do mapeamento DE-RX:

Com base no nosso mapeamento DE-RX, encontramos as seguintes regras funcionais:

- a. **nome, sobrenome, num_tel** → data_nasc, endereco, sexo, email, senha, eh_locatario, eh_anfitriao, id_local.
- b. **id_local** → país, estado, cidade, bairro.
- c. **nome, endereco, nomeUsu, sobrenomeUsu, num_telUsu** → tipo, qtd_quartos, qtd_banheiros, max_hospedes, min_noites, max_noites, preco_noite, taxa_limpeza, datas_diponivel, horario_checkIn, horario_checkOut, id_local.
- d. **id_quarto, nomeProp, endProp** → qtd_camras, tipo_cama
- e. **num_conta, nomeUsu, sobrenomeUsu, num_telUsu** → num_roteamento, tipo_conta).
- f. **tempo, nomeUsu, sobrenomeUsu, num_telUsu, nomeProp, endProp** → data_checkIn, fotos, rate_limpeza, rate_comunicacao, rate_localizacao, mensagem.
- g. **data_checkIn, nomeUsu, sobrenomeUsu, num_telUsu, nomeProp, endProp** → data_checkOut, data_reserva, qtd_hospedes, imposto, preco_estadia, cod_promocional, desconto, preco_total, condicao.
- h. **ponto, id_local** → nada
- i. **comodidade, nomeProp, endProp** → nada
- j. **regra, nomeProp, endProp** → nada

No entanto, encontramos muitas dificuldades para conseguir utilizar essas regras efetivamente, como podemos ver no arquivo Equivalências.pdf, a grande maioria dos dados que tivemos em nosso mapeamento não tinha equivalência com o airbnb. Por isso, embora saibamos ser errado, optamos por normalizar as tabelas pelos dados aos quais tínhamos acesso do Airbnb.

2. Tabelas cruas:

Pelo o que analisamos das tabela e conseguimos verificar depois com as equivalências feitas(no arquivo Equivalencias.pdf), identificamos o que mais se aproxima de chaves primárias (negrito e sublinhadas) e de chaves candidatas (apenas sublinhadas) de cada tabela. Elas são, respectivamente:

- Listings(**id**)
- Calendar(**listing_id**, **date**)
- Review(**id**, **listing_id**, **reviewer_id**, **date**)

Portanto, as tabelas cruas finais são:

- Listings(id, name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count, availability_365, number_of_reviews_ltm, license)
- Calendar(listing_id, date, available, price, adjusted_price, minimum_nights, maximum_nights)
- Review(id, listing_id*, date*, reviewer_id, reviewer_name, comments)
*listing_id e date juntos formam uma chave candidata.

A partir disso, utilizamos essas tabelas para criar as nossas regras funcionais, que serão utilizadas para normalizar:

- **id** → name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count, availability_365, number_of_reviews_ltm, license
- **listing_id, date** → available, price, adjusted_price, minimum_nights, maximum_nights
- **id** → listing_id, date, reviewer_id, reviewer_name, comments
- **listing_id, date** → reviewer_id, reviewer_name, comments

3. Primeira forma normal:

Para colocar uma relação na Primeira Forma Normal é necessário transformar os dados multivalorados em dados monovalorados. O único atributo multivalorado encontrado foi o **Listings(name)**, que decidimos dividir em outros 5 atributos (type_listing, mean_reviews, number_bedrooms, number_beds, number_bathrooms). Portanto, teremos as relações:

- Listings(id, type_listing, mean_reviews, number_bedrooms, number_beds, number_bathrooms, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count, availability_365, number_of_reviews_ltm, license)
- Calendar(listing_id, date, available, price, adjusted_price, minimum_nights, maximum_nights)
- Review(id, listing_id, date, reviewer_id, reviewer_name, comments)

4. Segunda forma normal:

Como as relações Listings e Review possuem somente uma chave, que contém um único atributo, e todos os outros atributos são dependentes deles, nada é mudado.

No entanto, a tabela Calendar possui uma só chave, mas que têm dois atributos nela (listing_id, date). Como os seus atributos 'price' e 'adjusted_price' dependem somente de parte da chave (listing_id), criaremos outra relação (Value). Pelo mesmo motivo, os atributos 'minimum_nights' e 'maximum_nights' deveriam criar uma nova tabela, mas como sua semântica seria muito parecida com a da tabela Listings, optamos por apenas adicionar tais atributos nessa velha relação. Assim, ficamos com as tabelas:

- Listings(id, type_listing, mean_reviews, number_bedrooms, number_beds, number_bathrooms, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, maximum_nights, number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count, availability_365, number_of_reviews_ltm, license)
- Calendar(listing_id, date, available)
- Value(listing_id, price, adjusted_price)
- Review(id, listing_id, date, reviewer_id, reviewer_name, comments)

5. Terceira forma normal:

A partir das relações da Primeira Forma Normal, tentaremos criar as da Terceira. Começando pela tabela Listings, vê-se que há alguns atributos, tais como 'host_name' e 'calculated_host_listings_count' que referem-se ao proprietário do imóvel, que tem 'host_id' como identificador único. Dessa forma, criamos a tabela Hosts contendo em cada instância as informações de cada proprietário. Além disso, algumas informações dizem respeito à localização de cada imóvel (latitude, longitude, neighbourhood, neighbourhood_group), como a vizinhança/bairro de algum lugar pode ser identificado somente com as coordenadas geográficas do local, colocamos 'latitude' e 'longitude' como suas chaves. Como os demais atributos referem-se à cada propriedade, as deixamos na mesma tabela Listings.

Já olhando para a tabela Calendar, observa-se que os seus atributos 'price' e 'adjusted_price' dependem somente de parte da chave (listing_id), assim faremos outra relação (Value). No entanto, como o atributo 'available' depende da chave inteira, o deixamos na tabela Calendar.

Por fim, olharemos para a relação Reviews. Nela, vê-se que o atributo 'reviewer_name' depende única e exclusivamente do identificador 'reviewer_id'. Assim, criamos uma nova tabela Reviewers contendo esses atributos. Além disso, como o resto dos atributos são identificados pela chave da relação, os mantemos na mesma tabela. Após todas essas operações, terminamos com essas relações:

- Listings(id, type_listing, mean_reviews, number_bedrooms, number_beds, number_bathrooms, host_id, room_type, minimum_nights, maximum_nights, number_of_reviews, last_review, reviews_per_month, number_of_reviews_ltm, number_of_reviews_ltm, license, latitude, longitude, price)
- Hosts(host_id, host_name, calculated_host_listings_count)
- Location(latitude, longitude, neighbourhood, neighbourhood_group)
- Calendar(listing_id, date, available)
- Value(listing_id, price, adjusted_price)
- Review(id, listing_id, date, reviewer_id, comments)

- Reviewers(reviewer_id, reviewer_name)

6. Forma normal de Boyce-Codd:

Como as relações que derivam de Listings e Calendar (Listings, Hosts, Location, Calendar, Value), além da tabela Reviewers, só possuem uma chave candidata, a Forma Normal de Boyce-Codd se reduzirá à Terceira Forma Normal, portanto teremos as mesma relações.

Já na tabela Review possuímos duas chaves candidatas, sendo a primária Review(id) e a outra Review(listing_id, date, reviewer_id). Devido ao único atributo não chave dessa relação ser dependente de ambas as chaves inteiramente, não será necessário mudar nada ao comparar com a Terceira Forma Normal. Portanto, ficaremos com as mesma tabelas da Terceira Forma Normal, sendo elas:

- Listings(id, type_listing, mean_reviews, number_bedrooms, number_beds, number_bathrooms, host_id, room_type, minimum_nights, maximum_nights, number_of_reviews, last_review, reviews_per_month, number_of_reviews_ltm, number_of_reviews_ltm, license, latitude, longitude, price)
- Hosts(host_id, host_name, calculated_host_listings_count)
- Location(latitude, longitude, neighbourhood, neighbourhood_group)
- Calendar(listing_id, date, available)
- Value(listing_id, price, adjusted_price)
- Review(id, listing_id, date, reviewer_id, comments)
- Reviewers(reviewer_id, reviewer_name)