# Project I: Design and Implement a Relational Database

*Requirements and background*

We gathered listing data from Airbnb in San Francisco. We filtered and left the most related fields. We created UML class diagrams, ERD Crow foot notation, relational schemas, and generated data to create a relational database. We also perform SQL queries to learn about the properties of Airbnb rentals.

Rules:

Field business rules:

- There are 500 listings and 400 hosts in the database.
- Listings only exist in the following 29 districts in San Francisco, including Sunset District, Union Square, Haight-Ashbury, Telegraph Hill, Marina, Castro, Golden Gate Park, Chinatown, Fisherman Wharf, Financial District, Mission District, North Beach, Presidio, Tenderloin, Japantown, Embarcadero, Noe Valley, Nob Hill, Soma, Pacific Heights, Bayview, Mission Bay, Stonestown, Hayes Valley, Bernal Heights, Alamo Square, Lower Pacific Heights, Stonestown, Russian Hill.
- There are only 5 types of Host response time, including within an hour, within a few hours, N/A, a few days or more, within a day.
- There are 15 types of room property.
- 10 kinds of room type.

Relationship business rules.

- One listing has one room.
- Every host must have zero or more rooms. Each host has one response time, each host has a district.
- Many hosts can be in the same district, many neighborhoods can be in the same district.
- One neighborhood can have zero or more rooms and can have many listings.
- Each room matches one piece of review and has the price from $80 - $300 per night. There is also description of the bathroom and number of bedroom and beds. Many rooms can be of the same property type, many rooms can have the same room type.

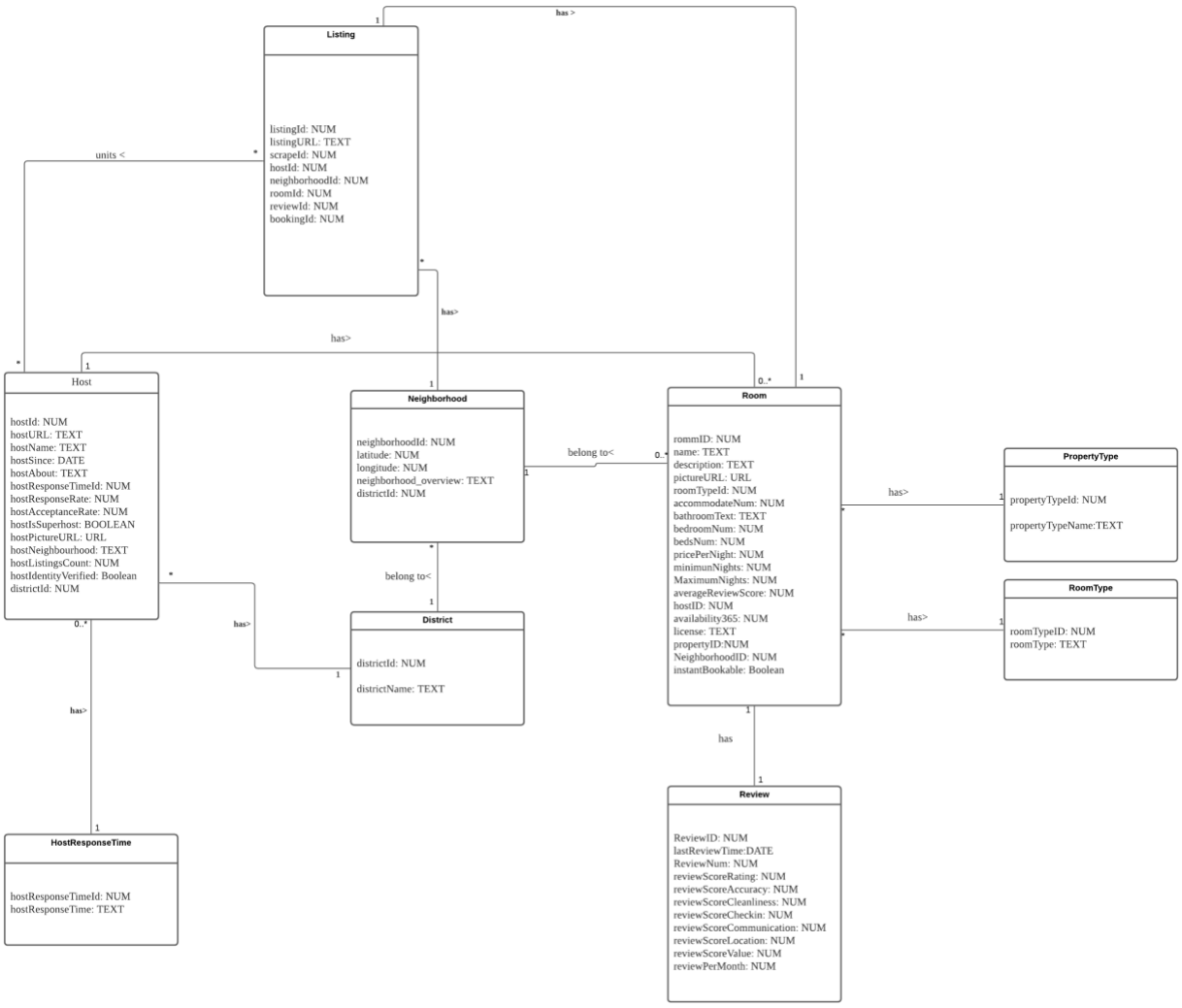
Nouns:

- Listing(Listings information include listing url, host, neighborhood, room, review)
- Host ( including information like hostURL, hostName ,hostSince, hostAbout,hostResponseTimeId, hostResponseRate, hostAcceptanceRate, hostIsSuperhost,hostPictureURL, hostNeighbourhood, hostListingsCount, hostIdentityVerified, districtId)
- HostResponseTime(including hostResponseTime)
- Neighborhood(including longitude,latitude,overview,district)
- District(including names of different districts in San Francisco)
- Unit(describes relationship between listings and host)
- Room(including the room description, price, minimum/maximum staying nights, review about the room, and its propertyType and room Type)
- PropertyType(like private room in rental unit)
- RoomType (private room, entire apt/home, entire loft)
- Review(including the review about the room cleanliness, check in process, overall value, description accuracy, communication)

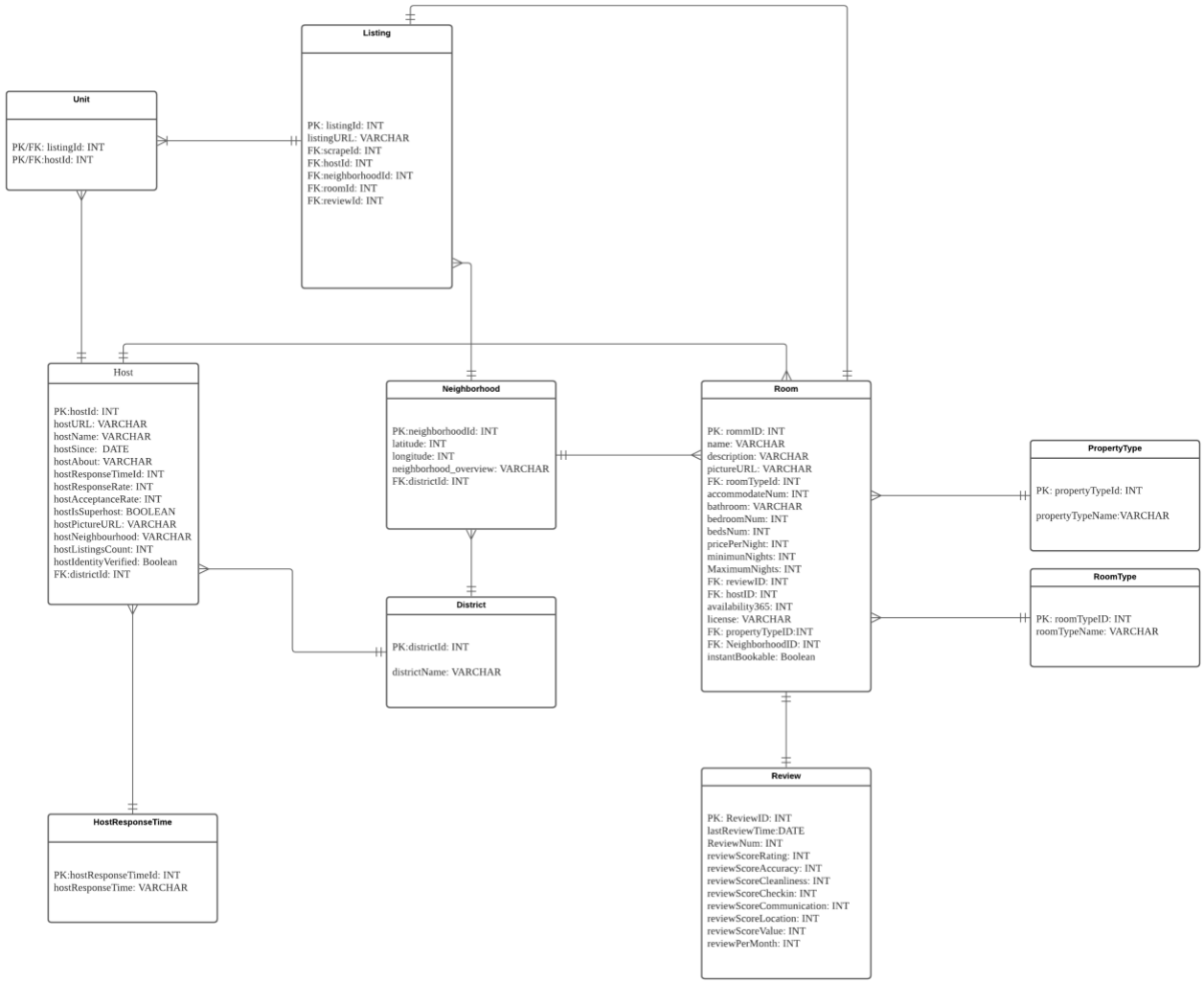
Actions:

- Has
- Belongs to
- Units

## *UML*


Listing
listingId: NUM
listingURL: TEXT
scrapeId: NUM
hostId: NUM
neighborhoodId: NUM
roomId: NUM
reviewId: NUM
bookingId: NUM
has >
units <
has>
Host
hostId: NUM
hostURL: TEXT
hostName: TEXT
hostSince: DATE
hostAbout: TEXT
hostResponseTimeId: NUM
hostResponseRate: NUM
hostAcceptanceRate: NUM
hostIsSuperhost: BOOLEAN
hostPictureURL: URL
hostNeighbourhood: TEXT
hostListingsCount: NUM
hostIdentityVerified: Boolean
districtId: NUM
Neighborhood
neighborhoodId: NUM
latitude: NUM
longitude: NUM
neighborhood_overview: TEXT
districtId: NUM
belong to<
Room
rommID: NUM
name: TEXT
description: TEXT
pictureURL: URL
roomTypeId: NUM
accommodateNum: NUM
bathroomText: TEXT
bedroomNum: NUM
bedsNum: NUM
pricePerNight: NUM
minimunNights: NUM
MaximumNights: NUM
averageReviewScore: NUM
hostID: NUM
availability365: NUM
license: TEXT
propertyID:NUM
NeighborhoodID: NUM
instantBookable: Boolean
PropertyType
propertyTypeId: NUM
propertyTypeName:TEXT
RoomType
roomTypeID: NUM
roomType: TEXT
District
districtId: NUM
districtName: TEXT
HostResponseTime
hostResponseTimeId: NUM
hostResponseTime: TEXT
has
Review
ReviewID: NUM
lastReviewTime:DATE
ReviewNum: NUM
reviewScoreRating: NUM
reviewScoreAccuracy: NUM
reviewScoreCleanliness: NUM
reviewScoreCheckin: NUM
reviewScoreCommunication: NUM
reviewScoreLocation: NUM
reviewScoreValue: NUM
reviewPerMonth: NUM

*ERD*




Relational Schemas:

Listing(listingId, listingURL, hostId, neighborhoodId, roomId, reviewId)

Host(hostId, hostURL, hostName ,hostSince, hostAbout,hostResponseTimeId, hostResponseRate, hostAcceptanceRate, hostIsSuperhost,hostPictureURL, hostNeighbourhood, hostListingsCount, hostIdentityVerified, districtId)

HostResponseTime(hostResponseTimeId, hostResponseTime)

Neighborhood(neighborhoodId, latitude, longitude, neighborhood_overview, districtId)

District(districtId, districtName)

Unit(listingId,hostId)

Room( rommID, name, description, pictureURL, roomTypeId, accommodateNum, bathroom, bedroomNum, bedsNum, pricePerNight, minimunNights, MaximumNights, averageReviewID, hostID, availability365, license, propertyID, NeighborhoodID, instantBookable)

Property(propertyID, propertyTypeName)

RoomType(roomTypeID, roomTypeName)

Review(ReviewID, lastReviewTime, ReviewNum, reviewScoreRating, reviewScoreAccuracy, reviewScoreCleanliness, reviewScoreCheckin, reviewScoreCommunication, reviewScoreLocation, reviewScoreValue, reviewPerMonth)

BCNF:

Listing Schema

Functional Dependency 1
listingId -> listingURL, hostId, neighborhoodId, roomId, reviewId (PK) Functional Dependency 2
listingId,listingURL->hostId, neighborhoodId, roomId, reviewId(Candidate key)

Functional Dependency 3
listingId, hostId -> listingURL, neighborhoodId, roomId, reviewId(Candidate key)
Functional Dependency 4
listingId,neighborhoodId -> listingURL, hostId, roomId, reviewId(Candidate key)
Functional Dependency 5
listingId,roomId -> listingURL, hostId, neighborhoodId, roomId, reviewId(Candidate key)
Functional Dependency 6
listingId,reviewId -> listingURL, hostId, neighborhoodId, roomId, reviewId(Candidate key)

So the Listing relation is in BCNF since every determinant is a candidate key.

Host Schema:

Functional Dependency 1
hostId -> hostURL ,hostName ,hostSince, hostAbout,hostResponseTimeId, hostResponseRate, hostAcceptanceRate, hostIsSuperhost,hostPictureURL, hostNeighbourhood, hostListingsCount, hostIdentityVerified, districtId(PK) Functional Dependency 2
hostId, hostURL -> hostName ,hostSince, hostAbout,hostResponseTimeId, hostResponseRate, hostAcceptanceRate, hostIsSuperhost,hostPictureURL, hostNeighbourhood, hostListingsCount, hostIdentityVerified, districtId(Candidate key)

Functional Dependency 3
hostId, hostName -> hostURL ,hostSince, hostAbout,hostResponseTimeId, hostResponseRate, hostAcceptanceRate, hostIsSuperhost,hostPictureURL, hostNeighbourhood, hostListingsCount, hostIdentityVerified, districtId(candidate key)

...
So host relation is in BCNF since every determinant is a candidate key.

HostResponseTime

Functional Dependency 1:
hostResponseTimeId -> hostResponseTime (PK)
So HostResponseTime relation is in BCNF since every determinant is a candidate key.

Neighborhood

Functional Dependency 1:

neighborhoodId -> latitude, longitude, neighborhood_overview, districtId (PK) Functional Dependency 2:
neighborhoodId, latitude -> longitude, neighborhood_overview, districtId (CK) Functional Dependency 3:

neighborhoodId, longitude-> latitude,, neighborhood_overview, districtId (CK) Functional Dependency 4:
neighborhoodId,neighborhood_overview->longitude, latitude, districtId (CK) Functional Dependency 5: neighborhoodId, districtId->neighborhood_overview->longitude, latitude(CK)
So Neighborhood relation is in BCNF since every determinant is a candidate key.

District

Functional Dependency 1:
districtId -> districtName (PK)
So District relation is in BCNF since every determinant is a candidate key.

Unit
Functional Dependency 1:
listingId -> hostId(PK)
So Unit relation is in BCNF since every determinant is a candidate key.

Room

Functional Dependency 1
rommID ->name, description, pictureURL, roomTypeId, accommodateNum, bathroom, bedroomNum, bedsNum, pricePerNight, minimunNights, MaximumNights, averageReviewID, hostID, availability365, license, propertyID, NeighborhoodID, instantBookable(PK)

Functional Dependency 2
rommID,name -> description, pictureURL, roomTypeId, accommodateNum,

bathroom, bedroomNum, bedsNum, pricePerNight, minimunNights, MaximumNights, averageReviewID, hostID, availability365, license, propertyID, NeighborhoodID, instantBookable(Candidate key)

Functional Dependency 3
rommID,description-> name, pictureURL, roomTypeId, accommodateNum,

bathroom, bedroomNum, bedsNum, pricePerNight, minimunNights, MaximumNights, averageReviewID, hostID, availability365, license, propertyID, NeighborhoodID, instantBookable(Candidate key)

...
So Room relation is in BCNF since every determinant is a candidate key.

Property

Functional Dependency 1:
propertyID -> propertyTypeName(PK)
So Property relation is in BCNF since every determinant is a candidate key.

RoomType

Functional Dependency 1:
roomTypeID -> roomTypeName(PK)
So RoomType relation is in BCNF since every determinant is a candidate key.

Review

Functional Dependency 1
ReviewID -> lastReviewTime, ReviewNum, reviewScoreRating, reviewScoreAccuracy, reviewScoreCleanliness, reviewScoreCheckin, reviewScoreCommunication, reviewScoreLocation, reviewScoreValue, reviewPerMonth(PK)

Functional Dependency 2
ReviewID, ReviewNum-> lastReviewTime,, reviewScoreRating, reviewScoreAccuracy, reviewScoreCleanliness, reviewScoreCheckin, reviewScoreCommunication, reviewScoreLocation, reviewScoreValue, reviewPerMonth(Candidate key)

Functional Dependency 3
ReviewID, ReviewNum-> lastReviewTime, reviewScoreRating, reviewScoreAccuracy, reviewScoreCleanliness, reviewScoreCheckin, reviewScoreCommunication, reviewScoreLocation, reviewScoreValue, reviewPerMonth(Candidate key)
...
So the Review relation is in BCNF since every determinant is a candidate key.