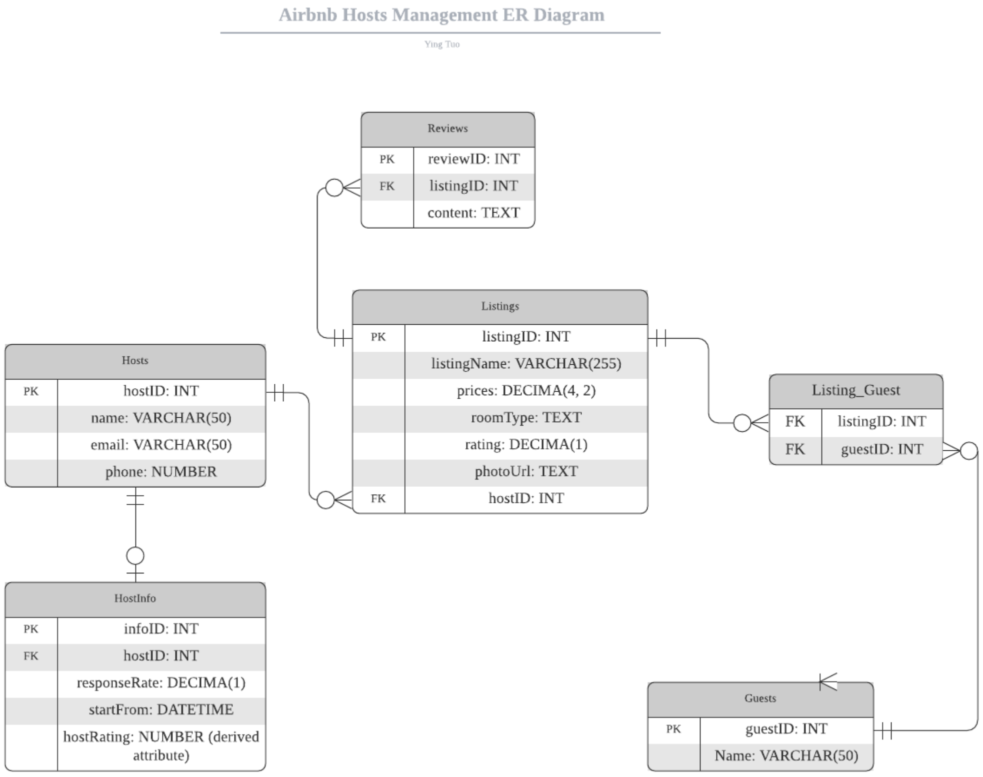
BCNF Proof

- (15 pts) From the logical model, define a relational schema in at least BCNF. Using functional dependencies, show that the schema in in at least BCNF.



Airbnb Hosts Management ER Diagram

Ying Tuo

The tables in our database are all normal form because they all follow these rules:

“The key” – (1NF) rule: There are no duplicate tuples in tables, which prevent a table from having a primary key. Every table has a primary key.
“The whole key” – (2NF): every attribute must depend on the entire primary key.
“Nothing but the key,” – (3NF): no dependencies on non-key attributes. No value depends on other values that aren't the primary key.
In our database, the hostRating comes from the rating in the Listing table, so it does not need to physically exist in the database, which obeys the 3NF rules.
“No more than one Candidate Key” – (BCNF): for a table to satisfy BCNF, other than table should be in the third normal form, it also should for a dependency A -> B, A should be a super key. Therefore, in my database schema, in the Hosts Table, hostID -> name, email, phone, in words we

can say name, email, phone is functionally dependent on hostID. Other tables have the same roles. For HostInfo Table, infoID -> responseRate, startFrom, hostRating; for Listings table, listingID -> hostID, listingName, prices, roomType, rating, photoUrl; for Reviews table, reviewID -> listingID, content; for the Guests table, guestid-> name. Therefore, there is no more than one candidate key

Overall, this logical model has a relational schema in BCNF.

1. Hosts Table

In this table, we have
<<key>> hostID: INT is our primary key, which is the unique id for Airbnb host;
name: VARCHAR(50) is the name of the Host;
email: VARCHAR(50)is the email of the host.
phone: NUMBER is the phone number of the host.

hostID	INT	Primary Key
name	VARCHAR(50)	
email	VARCHAR(50)	
phone	NUMBER	

2. HostInfo Table

In this table, we have
<<key>> infoID: INT is our primary key, which is the unique id for Airbnb host;
hostID: INT is the foreign for this host information;
responseRate: NUMBER is an evaluation criteria for the hosts. Airbnb staff can fill it based on other database;
startFrom: DATETIME denoted the date that the host joined Airbnb
hostRating: NUMBER (derived attribute). It is worthy to notice that this attribute is a derived attribute, which is not actually physically exists in our Airbnb database. This attribute is calculated based on the rating the Listings Table

infoID	INT	Primary Key
hostID	INT	Foreign Key
responseRate	DECIMA (1)	
startFrom	DATETIME	

3. Listing Table

<<key>> listingID: INT is the primary key for this table

hostID: INT is the foreign key for this listing. Noted, when we want to delete a host, the corresponding listings with hostid and reviews with listingid linked with this hosted should be deleted at the same time. Therefore, we set the contains for this foreign key when create Listings table as below,

FOREIGN KEY("hostid") REFERENCES "Hosts"("hostid") ***ON DELETE CASCADE***

listingName: VARCHAR(255)is the name of this listing

prices: DECIMA(4, 2) is the price/night for this listing

roomType: TEXT can be choose between Private Room and Entire Room;

rating: DECIMA(1)is the attribute that Airbnb staff to fill out based on the reviews of this listing;

photoUrl: TEXT can link to several photo profile of this listing.

listingID	INT	Primary Key
hostID	INT	Foreign Key
listingName	VARCHAR (255)	
prices	DECIMA (4,2)	
roomType	TEXT	
rating	DECIMA (1)	
photoUrl	TEXT	

4. Reviews Table

<<key>> reviewID: INT is the primary key for this table

listingID: INT is the foreign key for this review. Noted, when we want to delete a host, the corresponding listings with hostid and reviews with listingid linked with this hosted should be deleted at the same time. Therefore, we set the contains for this foreign key when create Reviews table as below,

FOREIGN KEY("listingid") REFERENCES "Listings"(" listingid ") ***ON DELETE CASCADE***

content: TEXT is the content of this review

reviewID	INT	Primary Key
listingID	INT	Foreign Key
content	TEXT	

5. Guests Table

<<key>> guestID: INT is the primary key for this table
listingID: INT builds the relationship between listings and guests. Since then have many to many relationship, so we need another table, Listing_Guest to build this relationship.
name: VARCHAR(50) is the name of guests

guestID	**INT**	Foreign Key
name	**VARCHAR(50)**	

6. Listing_Guests Table

listingID: INT holds the one to many relationship with Listings table
guestID: INT holds the one to many relationship with Guests table

listingID	**INT**	Foreign Key
guestID	**INT**	Foreign Key