Project Title: Airbnb Database Analysis & Application
Course No. & Section Number: CS 4347.002
Team Number: 34
Team Member(s): Abby Arce

Airbnb Database Analysis and Application

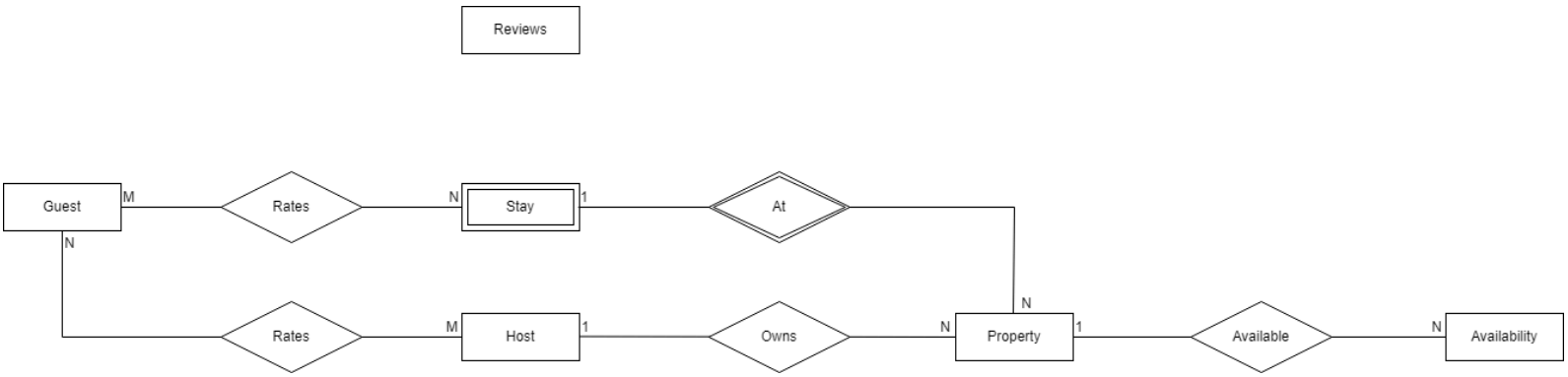
This project report will contain the contents required for this project. It includes a set up of the data requirements, as well as the ER diagrams, relational diagram, normalizations, and SQL commands done for this project.

Step 1: Data Requirements

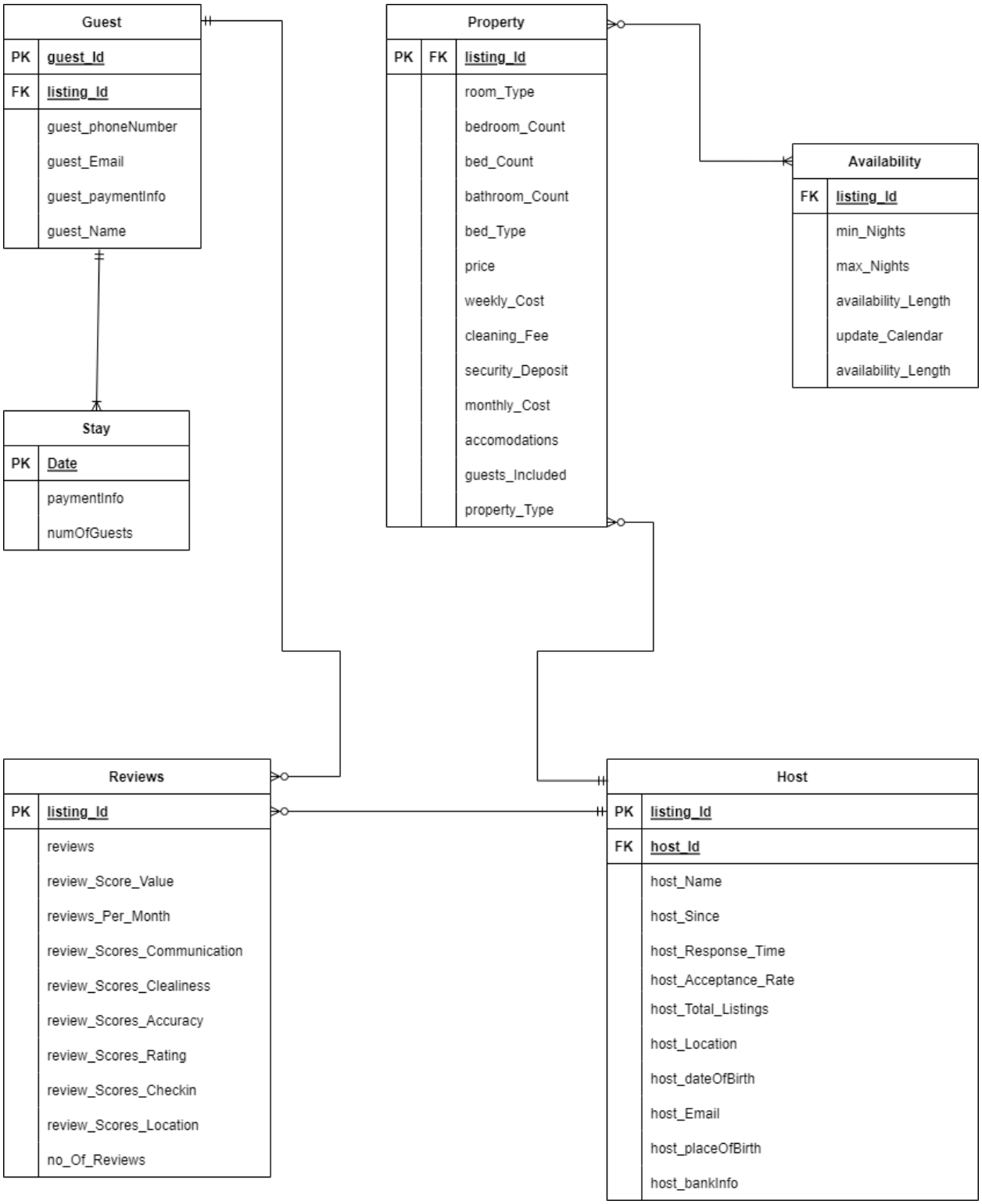
The following table shows the data requirements for this project:

Availability	Host	Property	Reviews	Guest	Stay		Interactions/Relationship
Listing_Id	Listing_Id	Listing_Id	Listing_Id	Listing_Id	Date		Guest-Rates-Stay
Min_Nights	Host_Name	Room_type	Review_Score_Value	guest_phoneNumber	paymentInfo		Host-Rates-Guest
Max_Nights	Host_Since	bedroom_Count	Reviews_Per_Month	guest_Email	numOfGuests		Host-Owns-Property
update_Calendar	Host_Response_Time	bed_Count	Review_Scores_Communication	guest_Id			Guest-Books-Stay
availability_Length	Host_Acceptance_Rate	bathroom_Count	Review_Scores_Clealiness	guest_paymentInfo			Stays-At-Property
	Host_Total_Listings	bed_Type	Review_Scores_Accuracy	guest_Name			
	Host_Location	Price	Review_Scores_Rating				
	Host_dateOfBirth	weekly_Cost	Review_Scores_Checkin				
	Host_Email	cleaning_Fee	Review_Scores_Location				
	Host_placeOfBirth	security_Deposit	No_Of_Reviews				
	Host_bankInfo	monthly_Cost					
	Host_Id	Accomodations					
		guests_Included					
		property_Type					

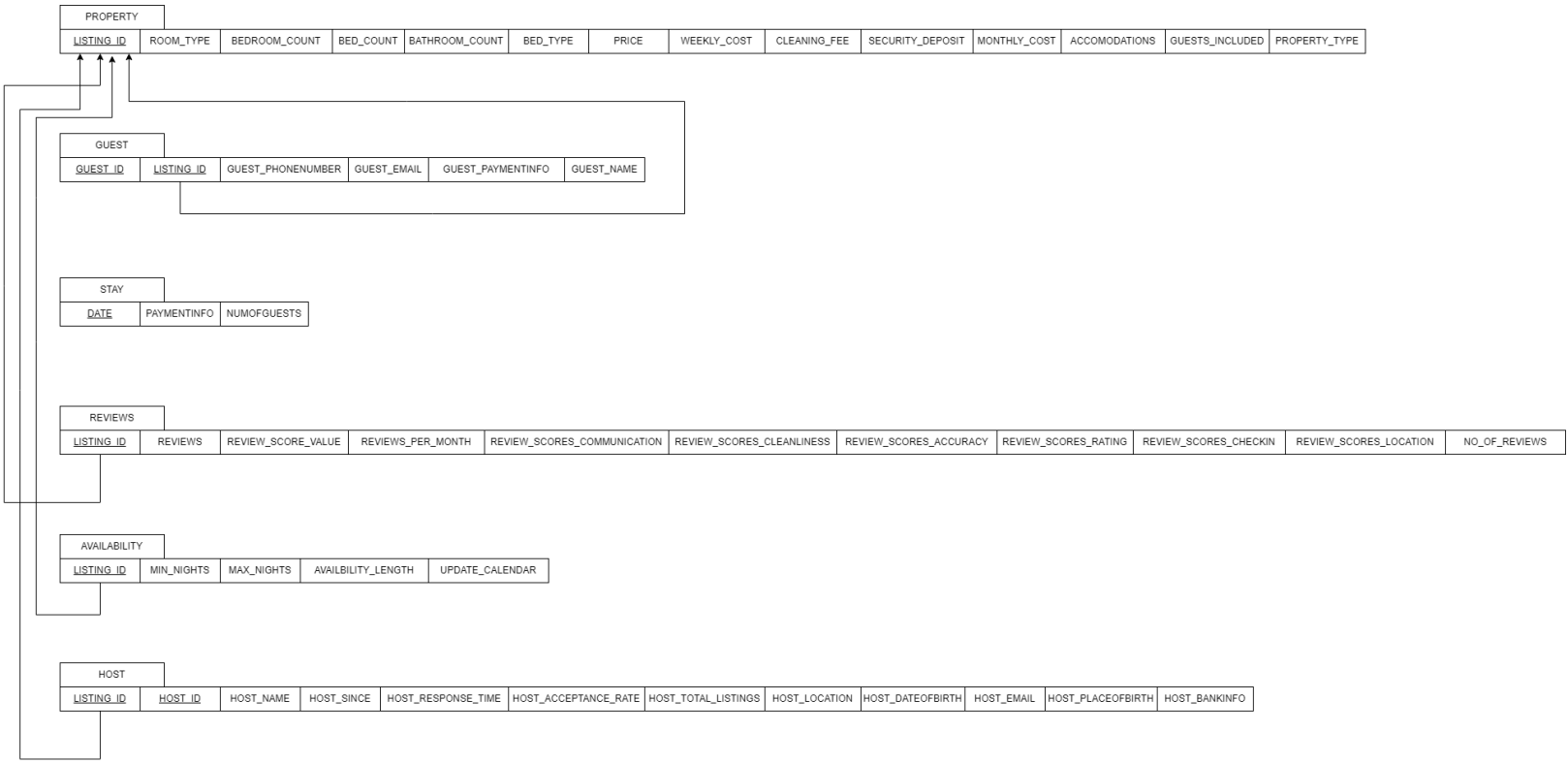
Step 2: ER Diagram



Another version shown below indicates the PK's and FK's



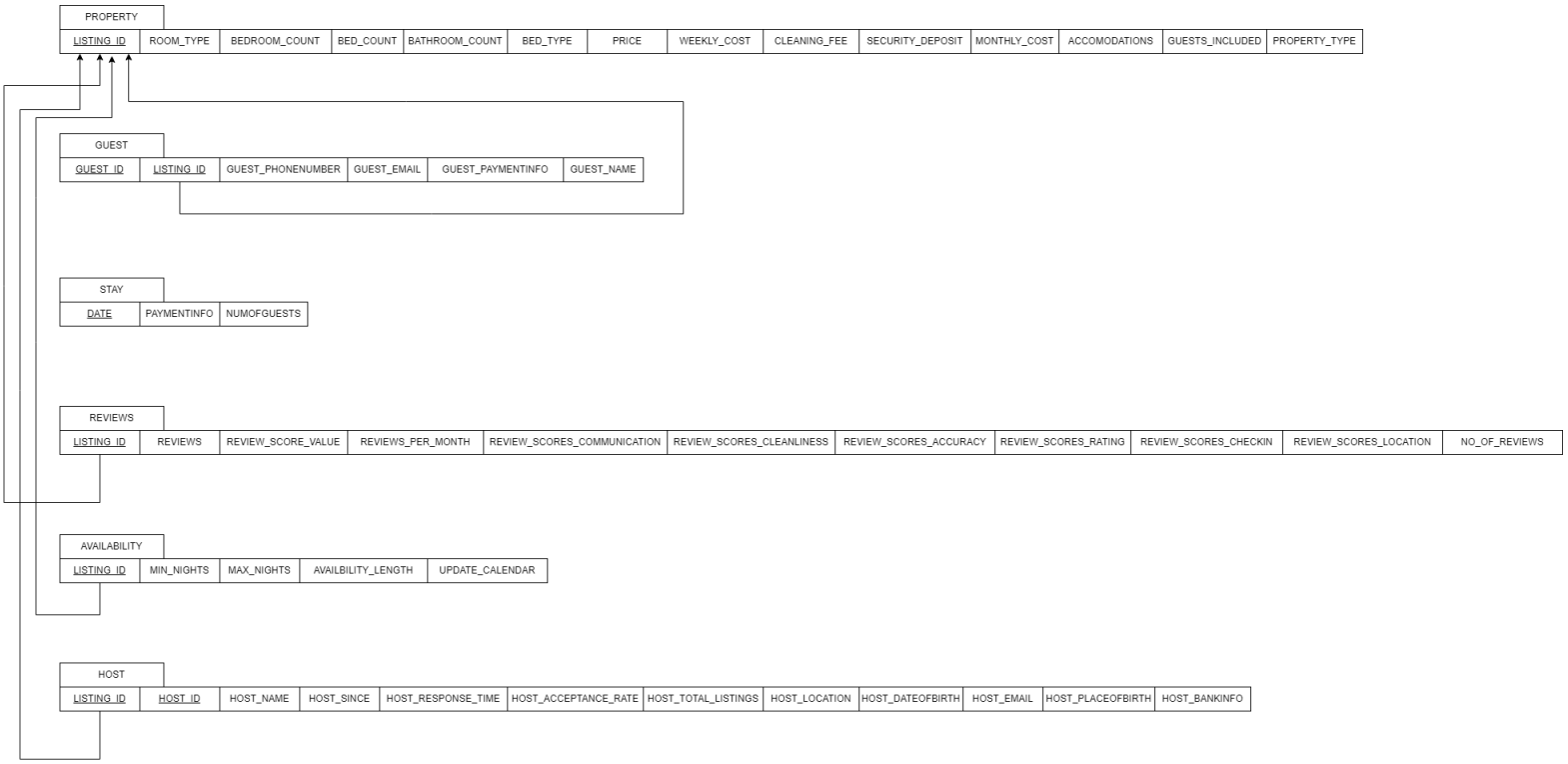
Step 3: ER Diagram to Relational Schema



Step 4: Normalization

For our tables, the functional dependencies seem to already be in 3NF. Therefore no changes needed to be made and the relational schema remains the same.

Step 5: Final Relational Schema (After Normalization)



Step 6: SQL Commands

```
CREATE TABLE Host {
listing_Id,
host_Name,
host_Since,
host_Response_Time,
host_Acceptance_Rate,
host_Total_Listings,
host_Location,
host_dateOfBirth,
host_Email,
host_placeOfBirth,
host_bankInfo,
host_Id,
PRIMARY KEY (host_Id),
FOREIGN KEY (listing_Id) REFERENCES Guest(listing_Id)
};
```

```
CREATE TABLE Property {
listing_Id,
room_type,
bedroom_Count,
bed_Count,
bathroom_Count,
bed_Type,
price,
weekly_Cost,
cleaning_Fee,
security_Deposit,
monthly_Cost,
accomodations,
guests_Included,
property_Type,
PRIMARY KEY (lisitng_Id)
};

CREATE TABLE Reviews {
listing_Id,
review_Score_Value,
reviews_Per_Month,
review_Scores_Communication,
review_Scores_Clealiness,
review_Scores_Accuracy,
review_Scores_Rating,
review_Scores_Checkin,
review_Scores_Location,
no_Of_Reviews,
FOREIGN KEY (listing_Id) REFERENCES Guest(listing_Id)

};

CREATE TABLE Guest {
listing_Id,
guest_phoneNumber,
guest_Email,
guest_Id,
guest_paymentInfo,
```

guest_Name,
PRIMARY KEY (guets_Id),
FOREIGN KEY (listing_Id) REFERENCES Guest(listing_Id)

};

CREATE TABLE Stay {
date ,
paymentInfo,
numOfGuests,
PRIMARY KEY (date)
};
[12:29 AM]
review_Score_Value
[12:29 AM]
review_Score_Value float
[12:30 AM]
review_Score_Value float NOT NULL
[12:30 AM]
date varchar(8)
[12:31 AM]
date varchar(10)