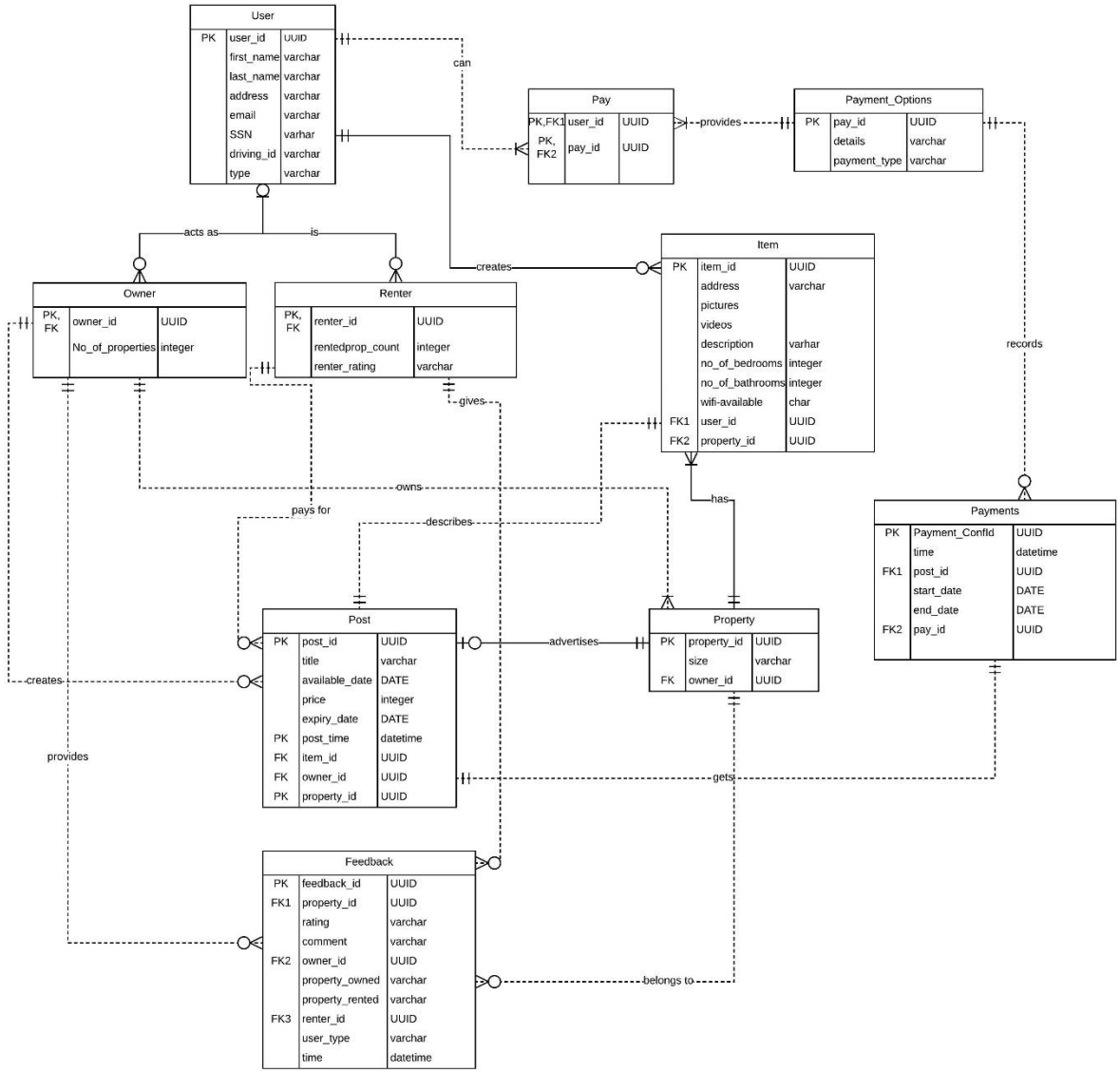
DATABASE SYSTEM CSCI 585

HomeWork1

SHome Report

Manish Dhankani

SHome ER Diagram



Components of ERM

ENTITY(X)	CARDINALITY(X)	RELATIONSHIP	ENTITY(Y)	CARDINALITY(Y)	CONNECTIVITY	DESCRIPTION
User	1,1	Acts as	Owner	0,M	1:M	A user can be owner of multiple house, but Owner would be a single user.
User	1,1	Acts as	Renter	0,M	1:M	A user can be owner of multiple house, but Renter would be a single user.
User	1,1	Can	Pay	1,M	1,M	A user can have multiple payments, but 1 payment belongs to 1 user only.
User	1,1	Creates	Item	0,M	1:M	A user can create multiple Items, but one item belongs to only 1 user.
Pay	1,M	provides	Payment_Options	1,1	M:1	Pay can be done via multiple payment options, but a payment option would have only 1 pay.
Owner	1,1	Owns	Property	1,M	1:M	An owner can own multiple properties, but one property belongs to only 1 owner.
Owner	1,1	Creates	Post	0,M	1:M	An owner can create multiple post, but one post belongs to only 1 owner.
Owner	1,1	Provides	Feedback	0,M	1:M	An owner can provide 0 or multiple

						feedback, but feedback
Renter	1,1	Pays for	Post	0,M	1:M	A Renter can pay for multiple posts, one post would be rented by 1 renter(who pays) at given time.
Renter	1,1	Gives	Feedback	0,M	1:M	A renter can give 0 or many feedback(s), and 1 feedback would be by only 1 renter.
Post	1,1	Describes	Item	1,1	1:1	A post is for one item, and an item belongs to only 1 post
Post	0,1	Advertises	Property	1,1	1:1	A post can advertise only one property, and there can be 1 property in 1 post.
Post	1,1	Gets	Payment	1,1	1:1	A post would receive 1 payment as entire payment needs to be made.
Feedback	0,M	Belongs to	Property	1,1	M:1	A single feedback belongs to 1 property, but one property can have multiple feedbacks.
Property	1,1	Has	Item	1,M	1:M	A property can have 1 or more items, but 1 item would belong to 1 property at a given time.
Payment_ options	1,1	Records	Payments	0,M	1:M	Payment can be done via multiple payment options, but 1 payment would have exactly 1 payment mode.(a single

						payment can't be shared via multiple options)

Assumptions:

1. Payments are made to SHome through which it gets credited to owner.
2. Only one user can be owner of property.
3. Only one user can be renter of property (means the one who makes payment, he/she can have multiple friends as renter(s)).
4. Renter must make entire payment at once using only one payment option.
5. The renter can't rent his own property would be handled by application or at implementation level.

Schema Definitions:

User

Attribute Name	Constraints
User_id	Primary key, Not null
First_name	Not null
Last_name	Not null
Address	Not null
Email	Not null
SSN	Not null
Driving_id	Not null
type	Not null

Description:

1. This entity is parent entity of Owner, Renter and Item entity.
2. It stores data specific to a user.
3. **User has complete overlapping as user can be Renter, Owner or both.**
4. Also, user must have at least one payment option.

Owner

Attribute Name	Constraints
Owner_id	Primary key, Not null, Foreign Key references user_id in User table
No_of_properties	Not null

Description:

1. This entity is weak entity and depends on User entity.
2. It stores data specific to an Owner.

Renter

Attribute Name	Constraints
User_id	Primary key, Not null, Foreign Key references user_id in User table
Renter_rating	Not null
Rentedprop_count	Not null

Description:

1. This entity is weak entity and depends on User entity.
2. It stores data specific to a renter.

Post

Attribute Name	Constraints
post_id	Primary key, Not null
title	Not null
Available_date	Not null
Price	Not null
Expiry_date	Not null
Post_time	Primary key, Not null
item_id	Foreign Key references item_id in Item table
Owner_id	Foreign Key references owner_id in Owner table
Property_id	Foreign Key references property_id in Property table

Description:

1. Title, available date, Price, Expiry_date and post_time cannot be null as all these attributes are responsible to provide complete information regarding property to user.

Property

Attribute Name	Constraints
Property_id	Primary key, Not null
Owner_id	Foreign key references owner_id in Owner table
size	Not null

Description:

1. Property_id, Owner_id and size cannot be null because if these are null, it won't provide correct information to user.

Pay

Attribute Name	Constraints

User_id	Primary key, Not null, Foreign key references user_id in User table
Pay_id	Primary key, Not null, Foreign key references pay_id in Payment_Options table

Description:

2. This entity acts as bridge entity between user and payment_options entity.
3. It stores primary keys of both user and payment_options entity.

Payment_Options

Attribute Name	Constraints
pay_id	Primary key, Not null
details	Not null
Payment_type	Not null

Description:

1. This entity stores data related to various payment options.
2. Pay_id, Details and payment_type cannot be null as there should be at least one (or more) payment option.

Payments

Attribute Name	Constraints
Payment_Confid	Primary key, Not null
time	Not null
Post_id	Foreign key references post_id in Post table
Start_date	Not null
End_date	Not null
Pay_id	Foreign Key references pay_id in Payment_options table

Description:

1. This entity records the payment with payment_confid at specific time and hence payment_confid cannot be null.
2. It relates to the type of payment using pay_id attribute.

Item

Attribute Name	Constraints
Item_id	Primary key, Not null
Address	Not null
Pictures	Not null
Videos	
Description	Not null
No_of_bedrooms	Not null

No_of_bathrooms	Not null
Wifi-available	Not null
User_id	Foreign key references user_id in User table
Property_id	Foreign Key references property_id in Property table

Description:

1. This entity is a weak entity and depends on User entity.
2. There should be at least 5 pictures and hence it is Not null, but the videos are optional and hence can be null.

Feedback

Attribute Name	Constraints
feedback_id	Primary key, Not null
Property_id	Foreign key references property_id in property table
Rating	
Comment	
Property_owned	Not null
Property_rented	Not null
Owner_id	Foreign key references owner_id in owner table
Renter_id	Foreign key references renter_id in renter table
Time	Not null

Description:

1. This entity stores feedback by both,
   a. Renter to property,
   b. Owner to Renter.
2. Also, Property_owned and property_rented would restrict only valid user(s) to provide rating and comments.
3. Rating, comment are not mandatory as the owner and renter may or may not provide rating and comments.

Design Choices:

1. To make the database more faster indexing can be done to primary keys of the tables.

Trade-offs with Other Designs:

1. To prevent many null values, I have tried to break large entities into smaller entities. But this would impact performance when we want to fetch data for large entity as we need to join the small tables(entities).