

## **Business Background:**

Rover is an online marketplace that creates a community for pet owners and trusted pet care providers. The founders strive to solve the problem of finding reliable, convenient pet care, offering services that include pet sitting, dog walking, pet daycare, and overnight boarding.

## **Assumptions:**

Pet Owner:

- Each pet owner has a unique email
- Payment info is stored securely for simple rebooking
- Referral system for pet owners to receive bonuses

Pet:

- Each pet has a primary owner
- Profile for pets that include detailed info and photos
- Specific care requirements and medical details

Pet Sitter:

- Background check for sitters
- Sitters can create their own availability and covered radius
- Sitters can offer multiple services with different price features

Booking:

- A booking is for one specific service
- Multiple pets can be booked from the same owner
- Bookings can be tracked (pending, confirmed, completed, cancelled)

## **Entity-Relationship Model (ER)**

Entities:

- PetOwner – Customer booking pet care
- Pet – Pets belonging to owners
- PetSitter – Provider offering pet care
- ServiceType – Service offerings (walking,sitting,boarding)
- Booking – Pet owners' reservation
- Review – Feedback for pet sitter

## Relationships:

- Owns (PetOwner 1:M Pet) – Each pet owner can have multiple pets
- Offers (PetSitter M:M ServiceTyper) – Sitters can offer multiple services with individual pricing
- Bookings (PetOwner M:M PetSitter through Booking) – Complex booking relationship
- Involves (Booking M:M Pet) – Bookings can include multiple pets
- Provides (Booking M:1 ServiceType) – Each booking is for one service type
- Reviews (PetOwner M:M PetSitter through Review) – Owner's review sitters

## Entity-Relationship Diagram (ERD)

### Initial Model:

PetOwners: (OwnerID, FirstName, LastName, Email, Phone, Address, City, State, ZipCode, JoinDate, ReferredByOwnerID(fk))

Pets: (PetID, OwnerID(fk), Name, Species, Breed, Age, Weight, Gender, MedicalNotes, PhotoURL, EmergencyContact)

PetSitters: (SitterID, FirstName, LastName, Email, Phone, Address, City, State, ZipCode, Bio, BackgroundCheckStatus, JoinDate, AvgRating)

ServiceTypes: (ServiceTypeID, ServiceName, Description, Category)

SitterServices: (SitterID(fk), ServiceTypeID(fk), PricePerHour, PricePerDay, PricePerWeek, Available, Description)

Bookings: (BookingID, OwnerID(fk), SitterID(fk), ServiceTypeID(fk), StartDateTime, EndDateTime, TotalPrice, Status, SpecialInstructions, CreatedDate, PaymentStatus)

PetBooking: (BookingID(fk), PetID(fk))

Reviews: (ReviewID, OwnerID(fk), SitterID(fk), BookingID(fk), Rating, Comment, ReviewDate, )

### Functional Dependencies Analysis:

#### PetOwners:

- FD1: OwnerID → FirstName, LastName, Email, Phone, Address, City, State, ZipCode, JoinDate, ReferredByOwnerID

#### Pets:

- FD1: PetID → OwnerID, Name, Species, Breed, Age, Weight, Gender, MedicalNotes, PhotoURL, EmergencyContact

PetSitters:

- FD1: SitterID → FirstName, LastName, Email, Phone, Address, City, State, ZipCode, Bio, BackgroundCheckStatus, JoinDate, AvgRating

ServiceTypes:

- FD1: ServiceTypeID → ServiceName, Description, Category

SitterServices:

- FD1: SitterID, ServiceTypeID → HourlyPrice, DailyPrice, Available, Description

Bookings:

- FD1: BookingID → OwnerID, SitterID, ServiceTypeID, StartDateTime, EndDateTime, TotalPrice, Status, SpecialInstructions, CreatedDate, PaymentStatus

PetBookings:

- FD1: BookingID, PetID → (no non-key attributes)

Reviews:

- FD1: ReviewID → OwnerID, SitterID, BookingID, Rating, Comment, ReviewDate, HelpfulVotes
- FD2: BookingID → OwnerID, SitterID (derived from Bookings table)

## Normalization to 3NF

Analysis: All relations are already in 3NF because:

- **1NF:** All attributes contain atomic values, no repeating groups
- **2NF:** No partial dependencies exist (all non-key attributes fully depend on entire primary key)
- **3NF:** No transitive dependencies exist (no non-key attribute depends on another non-key attribute)

The only potential concern was AvgRating in PetSitters, which could be considered derived data, but it's kept for performance reasons as it's frequently accessed.

## Final Relational Model (3NF)

PetOwners (OwnerID, FirstName, LastName, Email, Phone, Address, City, State, ZipCode, JoinDate, ReferredByOwnerID(fk))

- Primary Key: OwnerID
- Foreign Key: ReferredByOwnerID references PetOwners(OwnerID)

Pets (PetID, OwnerID(fk), Name, Species, Breed, Age, Weight, Gender, MedicalNotes, PhotoURL, EmergencyContact)

- Primary Key: PetID
- Foreign Key: OwnerID references PetOwners(OwnerID)

PetSitters (SitterID, FirstName, LastName, Email, Phone, Address, City, State, ZipCode, Bio, BackgroundCheckStatus, JoinDate, AvgRating)

- Primary Key: SitterID

ServiceTypes (ServiceTypeID, ServiceName, BaseDescription, Category)

- Primary Key: ServiceTypeID

SitterServices (SitterID(fk), ServiceTypeID(fk), PricePerHour, PricePerDay, PricePerWeek, Available, Description)

- Primary Key: (SitterID, ServiceTypeID)
- Foreign Keys: SitterID references PetSitters(SitterID), ServiceTypeID references ServiceTypes(ServiceTypeID)

Bookings (BookingID, OwnerID(fk), SitterID(fk), ServiceTypeID(fk), StartDateTime, EndDateTime, TotalPrice, Status, SpecialInstructions, CreatedDate, PaymentStatus)

- Primary Key: BookingID
- Foreign Keys: OwnerID references PetOwners(OwnerID), SitterID references PetSitters(SitterID), ServiceTypeID references ServiceTypes(ServiceTypeID)

PetBookings (BookingID(fk), PetID(fk))

- Primary Key: (BookingID, PetID)
- Foreign Keys: BookingID references Bookings(BookingID), PetID references Pets(PetID)

Reviews (ReviewID, OwnerID(fk), SitterID(fk), BookingID(fk), Rating, Comment, ReviewDate, HelpfulVotes)

- Primary Key: ReviewID
- Foreign Keys: OwnerID references PetOwners(OwnerID), SitterID references PetSitters(SitterID), BookingID references Bookings(BookingID)

## Visual of ERD

