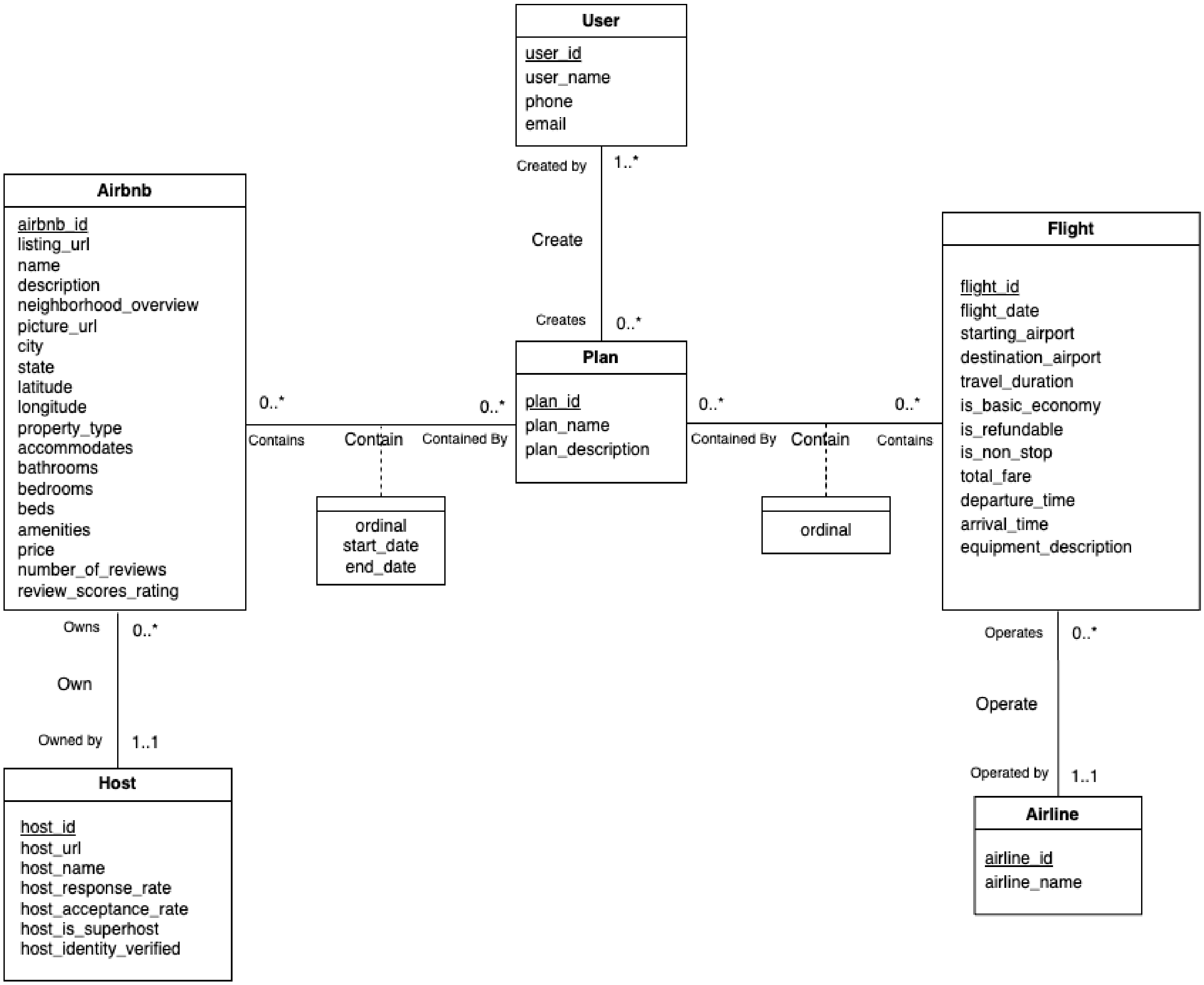
Travel Planner UML and Normalization:

UML diagram:



Explanation for Classes:

1. **Airbnb**: Represents a unique listing available for short-term rental.
   Reason: An Airbnb listing has multiple attributes like city, price, and reviews, making it a distinct class with its own properties and relationships (e.g., with hosts and plans).

2. **Host**: Refers to the individual that owns or manages one or more Airbnb listings. *Reason*: A host can manage multiple listings and have several non-key attributes requiring it to be a separate class with relationships to other classes like Airbnb listings.

3. **Flight**: Represents a scheduled air travel service between two locations. *Reason*: A flight has its own attributes like departure time, flight id, and destination, and it is associated with other classes such as airlines and plans, making it a standalone class.

4. **Airline**: The company responsible for operating flights. *Reason*: Having Airline as an attribute for Flight introduces unnecessary redundancy in the database. Any changes to airline details would require updates in multiple places. Hence, it is a separate class.

5. **Plan**: Refers to a travel or accommodation package that can include flights, Airbnbs, or both. *Reason*: A plan is a separate class because it aggregates multiple components (Flights, Airbnbs) and has its own attributes like plan id, plan name and plan description.

6. **User**: Represents the end-user of the application who creates plans. *Reason*: A user is a class because they have distinct attributes such as user id, phone, and email, and they establish relationships with the plan class.

Explanation for Relationship Cardinalities:

1. **Airbnb to Host**
   Reason: A host can own multiple Airbnb listings, but each listing is owned by only one host.

2. **Flight to Airline**
   Reason: An airline can operate multiple flights, but each flight is operated by only one airline.

3. **Plan to Airbnb**
   Reason: A plan can contain zero or more Airbnb listings, and an Airbnb can be part of multiple plans (or none at all).

4. **Plan to Flight**
   Reason: A plan can include zero or more flights, and a flight can be part of multiple plans (or none at all).

5. **User to Plan**
   Reason: A user can create multiple plans, and a plan may be shared by multiple users.

Converting UML to Relational Schema:

1. User(user_id, user_name, phone, email)

   User class is converted to User table. user_id is the primary key.

2. Airbnb(airbnb_id, listing_url, name, description, neighborhood_overview, picture_url, city, state, latitude, longitude, property_type, accommodates, bathrooms, bedrooms, beds, amenities, price, number_of_reviews, review_scores_rating, host_id)

   Airbnb class is converted to Airbnb table. airbnb_id is the primary key. host_id is added as a foreign key to the Host table.

3. Flight(flight_id, flight_date, starting_airport, destination_airport, travel_duration, is_basic_economy, is_refundable, is_non_stop, total_fare, departure_time, arrival_time, equipment_description, airline_id)

   Flight class is converted to Flight table. flight_id is the primary key. airline_id is added as a foreign key to the Airline table.

4. Host(host_id, host_url, host_name, host_response_rate, host_acceptance_rate, host_is_superhost, host_identity_verified)

   Host class is converted to Host table. host_id is the primary key.

5. Airline(airline_id, airline_name)

   Airline class is converted to Airline table. airline_id is the primary key.

6. Plan(plan_id, plan_name, plan_description)

   Plan class is converted to Plan table. plan_id is the primary key.

7. User_to_Plan(user_id, plan_id)

   User to Plan many-to-many relationship is converted to User_to_Plan table. (user_id, plan_id) is the primary key.

8. Plan_to_Flight(plan_id, flight_id, ordinal)

   Plan to Flight many-to-many relationship is converted to Plan_to_Flight table. (plan_id, flight_id) is the primary key as it uniquely determines the ordinal (ordinal refers to which leg of the plan the flight is)

9. Plan_to_Airbnb(plan_id, airbnb_id, ordinal, start_date, end_date)

   Plan to Airbnb many-to-many relationship is converted to Plan_to_Airbnb table. (plan_id, airbnb_id, ordinal) is the primary key as it uniquely identifies everything else in the table. We also include ordinal in the primary key because the same airbnb can be used in the plan again.

Identify FDs, prove that relations are in 3NF:

A relation is in 3NF if for all FDs X → Y, X is a super key or Y is part of some candidate key

1. User
   a. user_id → user_name, phone, email
   b. phone → user_id, user_name, email
   c. email → user_id, user_name, phone

   user_id, phone and email are candidate keys; user_id is the chosen primary key

   a. user_id is a super key
   b. phone is a super key
   c. email is a super key

   Hence, User is in 3NF

2. Airbnb
   a. airbnb_id → listing_url, name, description, neighborhood_overview, picture_url, city, state, latitude, longitude, property_type, accommodates, bathrooms, bedrooms, beds, amenities, price, number_of_reviews, review_scores_rating, host_id
   b. listing_url → airbnb_id

   airbnb_id and listing_url are candidate keys; airbnb_id is the chosen primary key

   a. airbnb_id is a super key
   b. listing_url is a super key

   Hence, Airbnb is in 3NF

3. Flight

a. flight_id → flight_date, starting_airport, destination_airport, travel_duration, is_basic_economy, is_refundable, is_non_stop, total_fare, departure_time, arrival_time, equipment_description, airline_id

flight_id is the only candidate key and so it is the chosen primary key

a. flight_id is a super key

Hence, Flight is in 3NF

4. Host
    a. host_id → host_url, host_name, host_response_rate, host_acceptance_rate, host_is_superhost, host_identity_verified
    b. host_url → host_id

host_id, host_url are candidate keys; host_id is the chosen primary key

a. host_id is a super key
b. host_url is a super key

Hence, Host is in 3NF

5. Airline
    a. airline_id → airline_name
    b. airline_name → airline_id

airline_id, airline_name are candidate keys; airline_id is the chosen primary key

a. airline_id is a super key
b. airline_name is a super key

Hence, Airline is in 3NF

6. Plan
    a. plan_id → plan_name, plan_description

plan_id is the only candidate key and so it is the chosen primary key

a. plan_id is a super key

Hence, Plan is in 3NF

7. User_to_plan

No FDs

As there are no FDs, User_to_plan is in 3NF

8. Plan_to_Flight
    a. (plan_id, flight_id) → ordinal

(plan_id, flight_id) is the only candidate key, it is the chosen primary key

a. (plan_id, flight_id) is a super key

Hence, Plan_to_Flight is in 3NF

9. Plan_to_Airbnb
    a. (plan_id, airbnb_id, ordinal) → (start_date, end_date)
    b. (plan_id, airbnb_id, start_date, end_date) → ordinal

(plan_id, airbnb_id, ordinal), (plan_id, airbnb_id, start_date, end_date) are the candidate keys; (plan_id, airbnb_id, ordinal) is chosen as the primary key

a. (plan_id, airbnb_id, ordinal) is a super key
b. (plan_id, airbnb_id, start_date, end_date) is a super key

Hence, Plan_to_Flight is in 3NF

Relational Schema:

User(user_id:VARCHAR [PK], user_name:VARCHAR, phone:VARCHAR, email:VARCHAR)

Airbnb(airbnb_id:VARCHAR [PK], listing_url:VARCHAR, nameVARCHAR, descriptionVARCHAR, neighborhood_overviewVARCHAR, picture_url:VARCHAR, city:VARCHAR, state:VARCHAR, latitude:DECIMAL(10, 7), longitude:DECIMAL(10, 7), property_type:VARCHAR, accommodates:INT, bathrooms:INT, bedrooms:INT, beds:INT, amenities:VARCHAR, price:(5, 3), number_of_reviews:INT, review_scores_rating:DECIMAL(1,1), host_id:VARCHAR [FK to Host.host_id])

Flight(flight_id:VARCHAR [PK], flight_date:DATE, starting_airport:VARCHAR, destination_airport:VARCHAR, travel_duration:DECIMAL(3, 2), is_basic_economy:BOOLEAN, is_refundable:BOOLEAN, is_non_stop:BOOLEAN, total_fare:DECIMAL(5, 2), departure_time:TIME, arrival_time:TIME, equipment_description:VARCHAR, airline_id:VARCHAR [FK to Airline.airline_id])

Host(host_id:VARCHAR [PK], host_url:VARCHAR, host_name:VARCHAR, host_response_rate:INT, host_acceptance_rate:INT, host_is_superhost:BOOLEAN, host_identity_verified:BOOLEAN)

Airline(airline_id:VARCHAR [PK], airline_name:VARCHAR)

Plan(plan_id:VARCHAR [PK], plan_name:VARCHAR, plan_description:VARCHAR)

User_to_Plan(user_id:VARCHAR [FK to User.user_id], plan_id:VARCHAR [FK to Plan.plan_id], [PK(user_id,plan_id)])

Plan_to_Flight(plan_id:VARCHAR [FK to Plan.plan_id], flight_id:VARCHAR [FK to Flight.flight_id], ordinal:INT, [PK(plan_id,flight_id)])

Plan_to_Airbnb(plan_id:VARCHAR [FK to Plan.plan_id], airbnb_id:VARCHAR [FK to Airbnb.airbnb_id], ordinal:INT, start_date:DATE, end_date:DATE, [PK(plan_id,airbnb_id,ordinal)])