Vacation Home Rental

Overview

Your assignment is to design a database system for a XYZ vacation home rental company. You will design the database, insert some sample data, and implement a set of required features. Each feature will be implemented as one or more Oracle PL/SQL procedures. You do **NOT** need to write a graphic user interface.

Assumptions:

You can make the following assumptions in this project.

1. The system needs to store data about users (you can assume that each user needs to register with the system). Users could be hosts or guests (it is possible for a user to be both). Each user has user ID (for internal use), name, mailing address (including street, city, state, and zipcode), phone number, email, and a password (email is used as username). In addition, a host has an average rating (computed from reviews for the host) and a method of receiving payment (you can assume it is bank account number as varchar). A guest has an average rating and a profile (as varchar) stating some information about the guest (e.g., gender, family, job, hobby, etc.).
2. The system also stores information about listings. Each listing has a host, an address, zip code, city, state, type (house or apartment), maximal capacity (number of people can live there), number of bedrooms, number of beds, number of bathrooms, minimal number of nights of stay, check in time and check out time (you can use interval data type).
3. Each listing has a list of amenities such as Wifi, TV, washer and dryer, free parking, etc.
4. A host can enter several availability periods for a listing. Each period as a start date, and end date (if guest check out on end date the price still applies, but if the guest check in on end date the price for next period will be used), and a price per night if a guest stay in this period.
   For example, suppose the availability periods are:
   Period 1: 2016-9-7 to 2017-6-1, price per night is $80;
   Period 2: 2017-6-1 to 2017-9-7, price per night is $110;
   Period 3: 2017-9-7 to 2018-6-1, price per night is $90.
   If a guest stay at this listing from 2017-9-1 (check in date) to 2017-9-10 (check out date). The guest stays 9 nights, 6 nights in period 2 and 3 nights in period 3. So the total charge of this stay is 6*110+3*90 = $930 plus a service fee (described next).
5. The rental company charges a 5% guest service fee for each booking and 3% host service fee. The guest service fee is added to a guest's charge per booking. The host service fee is deducted from the payment (called payout) received by hosts.
6. Each guest can enter review for a host after staying at a listing owned by the host. Each host can enter review for a guest as well.
7. The database stores booking information, including booking ID, guest ID, listing ID, check in date, check out date, number of adults, number of children, payment method (as varchar), payment date (normally one day before check in), booking status (requested, approved or denied, paid, canceled), and payout status (whether host receives payment from the rental company). The guest first submits a booking request. The host can approve or deny it. If the request is approved, the guest makes

payment to the rental company about one day before check in. The payment will be sent to host by the rental company (with guest and host service fee deducted) after some time.

8. A guest can cancel a booking before payment date. But not after that.
9. The system stores payout information (i.e., payment sent from rental company to host). The system usually computes total payment to a host once a month.
10. The system stores a message table, which includes user id, message date and message body.

Features:

Features for account management:

1. Register a user with the system. The customer needs to provide name, address, phone#, email, password, and user type (host or guest or both). The procedure should check whether the email already exists in user table. If so, please print a message saying the customer exists. Otherwise create an account with input values and return a new user ID.

2. Allow a user to login by providing email and password. Please check whether email exists and password matches. If not, please print a message to indicate the error. Otherwise print a message to indicate user has logged on. The procedure should return a value 1 for success login and 0 for unsuccessful log in.

3. Allow a user to read messages providing user id and a starting date. Print out messages for that user since that date.

Features for listing management

4. Allow a host to add a listing. The input includes host ID, address of the listing (street, city, state, zip), type of hosting, maximal capacity, number of bedrooms, beds, bathrooms, minimal stay, check in time and check out time, and a list of amenities (you can use varray data type). A new listing ID should be generated.

5. Allow a host to enter an availability period for a listing. The input includes listing ID, start date, end date, price per night.

Booking

6. ***(most difficult feature)
   Look up available houses at a given city and state and in a given period. Input: city and state, check in date and check out date. Output: display house address, total price for listings that are available in the entire period and there is no booking (do not count those denied or canceled bookings) on the same house overlap with that period. The total price is computed by sum of per_night_price for every day in the period pluse a 5% service charge by the company.

   Note that the given period (check in date to check out date) may overlap with multiple availability periods in listing_availability table. For example, suppose the given period is 2017-9-1 to 2017-9-15, but there are two available period for a listing: one from 2007-

6-1 to 2017-9-7 with cost per night 180 and the other from 2017-9-7 to 2018-6-1 with cost per night 100.

There are 6 nights in the first period with cost $180 per night and 8 in the second period with cost $100 per night.
So total cost = (180*6 + 100*8)*1.05 = 1974.
However, if the two available period is from 2007-6-1 to 2017-9-7 and from 2017-9-10 to 2018-6-1, the given period is not completely covered (2017-9-8 to 2017-9-9), then this listing should not be in the result.

Hint: You can first fetch all listings that are the given city and state and do not have reservations intersect with given period. For each such listing, you can call a sub procedure to check whether the given period is completely covered by one or more available periods and compute the total price.

7. Booking request: a guest sends a booking request with listing ID, guest ID, checkin date, checkout date, number of adults, number of kids.

The feature automatically checks whether the listing is available (see feature 6 for how to check availability), and whether it meets maximal capacity and min stay requirement. Generate a message for the guest if any of these conditions are not satisfied. If all conditions are satisfied, insert a row into booking table with status = 'requested' and generate a message for the host if all conditions are satisfied. The message contains the request information along with average review rating of the guest.

8. Allow a host to approve or deny a booking request. Input: booking id, decision (approval or deny). Output: check if the booking exists. If not print out an error message. Otherwise update the booking status (set it to approved or denied) and insert a message to both host and guest indicating the request has been approved or denied.

9. Look up booking request for a host. Input: host id. Output: booking id, guest name, listing id, check in date, check out date, number of guests for listings owned by that host with status = requested.

10. Allow a guest to make payment. Input includes booking ID, payment method, and payment date. Check whether the booking exists and is approved and the payment date is at least one day before checkin date. If so, update payment status to paid and insert a message for the host and guest about this payment. Otherwise, print an error message.

11. Allow a guest to cancel a booking if not paid yet. Input includes booking ID. If the booking does not exist or the status is paid, print an error message and insert a message to guest saying why it cannot be canceled. Otherwise update status to canceled and insert a message for guest and host about the cancelation.

12. Allow the system to generate payout to host. Input includes host ID, payout date. Find all bookings that are paid but with payout status be 0, update status to 1 (paid out) and compute total amount (sum of total for each booking/1.05 * 0.97 to exclude service fees). Insert a payout record to payout table. Insert a message for the host.

13. Allow a guest to enter a review for host and update average rating for the host as well.

Please check the guest indeed has a booking (paid) with a listing owned by the host.

14. Allow a host to enter a review to guest and update average rating for the guest as well. Please check the guest indeed has a booking (paid) with a listing owned by the host.

Analysis

15. Report the following statistics: total number of users, hosts, guests, listings, bookings, top-k hosts with highest average ratings (k as input), top-k guests with highest verage ratings, average length of stay per booking, average cost per booking.
    1.